US006178205B1

(12) **United States Patent**
Cheung et al.

(10) **Patent No.: US 6,178,205 B1**
(45) **Date of Patent: Jan. 23, 2001**

(54) **VIDEO POSTFILTERING WITH MOTION-COMPENSATED TEMPORAL FILTERING AND/OR SPATIAL-ADAPTIVE FILTERING**

(75) Inventors: **Sen-ching S. Cheung**, Fremont; **David Drizen**, San Jose; **Paul E. Haskell**, Saratoga, all of CA (US)

(73) Assignee: **VTEL Corporation**, Sunnyvale, CA (US)

(*) Notice: Under 35 U.S.C. 154(b), the term of this patent shall be extended for 0 days.

(21) Appl. No.: **08/989,839**

(22) Filed: **Dec. 12, 1997**

(51) **Int. Cl.**[7] **H04B 1/66**
(52) **U.S. Cl.** **375/240.29**
(58) **Field of Search** 348/845, 606, 348/607, 620, 699, 701, 403, 405, 413, 416, 667, 420, 448; 382/261, 263, 264, 272; 370/290; 375/240, 240.29; A04B 1/66

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,475,434	*	12/1995	Kim	348/420
5,481,628	*	1/1996	Ghaderi	382/261
5,502,489	*	3/1996	Kim et al.	348/607
5,539,469	*	7/1996	Jung	348/413
5,598,213	*	1/1997	Chung et al.	348/405
5,610,729	*	3/1997	Nakajima	348/607
5,621,468	*	4/1997	Kim	348/416
5,654,759	*	8/1997	Augenbraun et al.	348/405
5,742,344	*	4/1998	Odaka et al.	348/416
5,793,435	*	8/1998	Ward et al.	348/448
5,907,370	*	5/1999	Suzuki et al.	348/607

OTHER PUBLICATIONS

Liu, et al., "Adaptive Postprocessing Algorithms for Low Bit Rate Video Signals", *IEEE Transactions on Image Processing*, 4:7:1032–1035 (Jul., 1995).

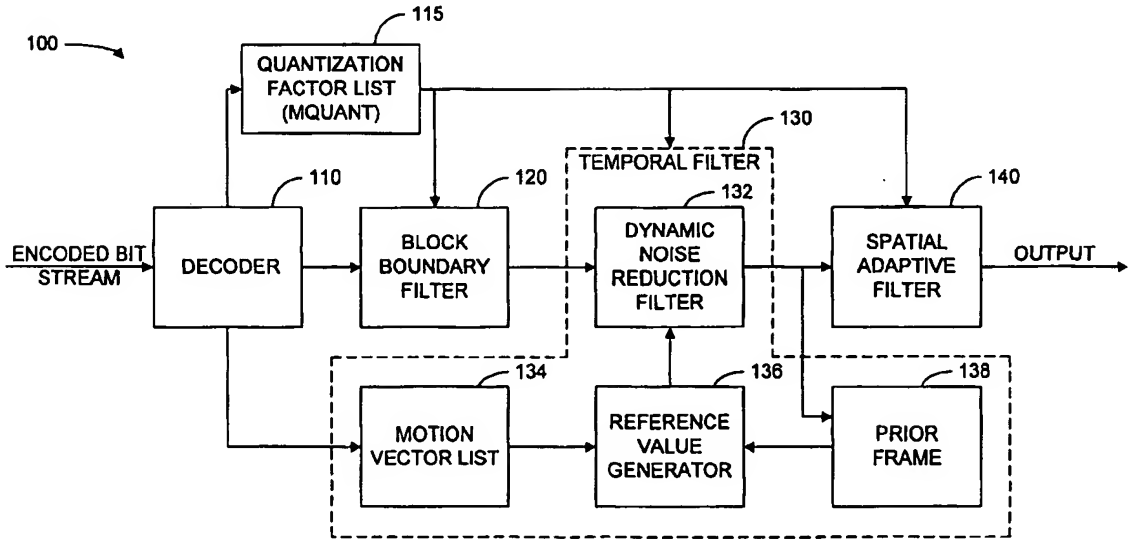
* cited by examiner

Primary Examiner—Howard Britton
Assistant Examiner—Gims Philippe
(74) *Attorney, Agent, or Firm*—Skjerven Morrill MacPherson LLP; David T. Millers

(57) **ABSTRACT**

A postfiltering process for improving the appearance of a video image includes motion compensated temporal filtering and spatial adaptive filtering. For each target pixel being filtered, the temporal filtering uses multiple motion vectors and one or more pixel values for a prior frame to determine one of more reference values for the target filter. In one embodiment, a weighted average of multiple motion vectors for blocks near or containing the target pixel value provides a filter vector that points to a pixel value in the prior frame. This pixel value is a reference value for the target pixel value and is combined with the target pixel value in a filter operation. Alternatively, multiple motion vectors for blocks near or containing the target pixel value point to pixel values in the prior frame that are averaged to determine a reference value for the target pixel value. In each alternative, the weighting for the average is selected according to the position of the target pixel value. The spatial filtering determines a dynamic range of pixel values in a smaller block containing the target pixel value and a dynamic range of pixel values in a larger block containing the target pixel value. The two dynamic ranges suggest the image context of the target pixel, and an appropriate spatial filter for the target pixel is selected according to the suggested context.

31 Claims, 5 Drawing Sheets



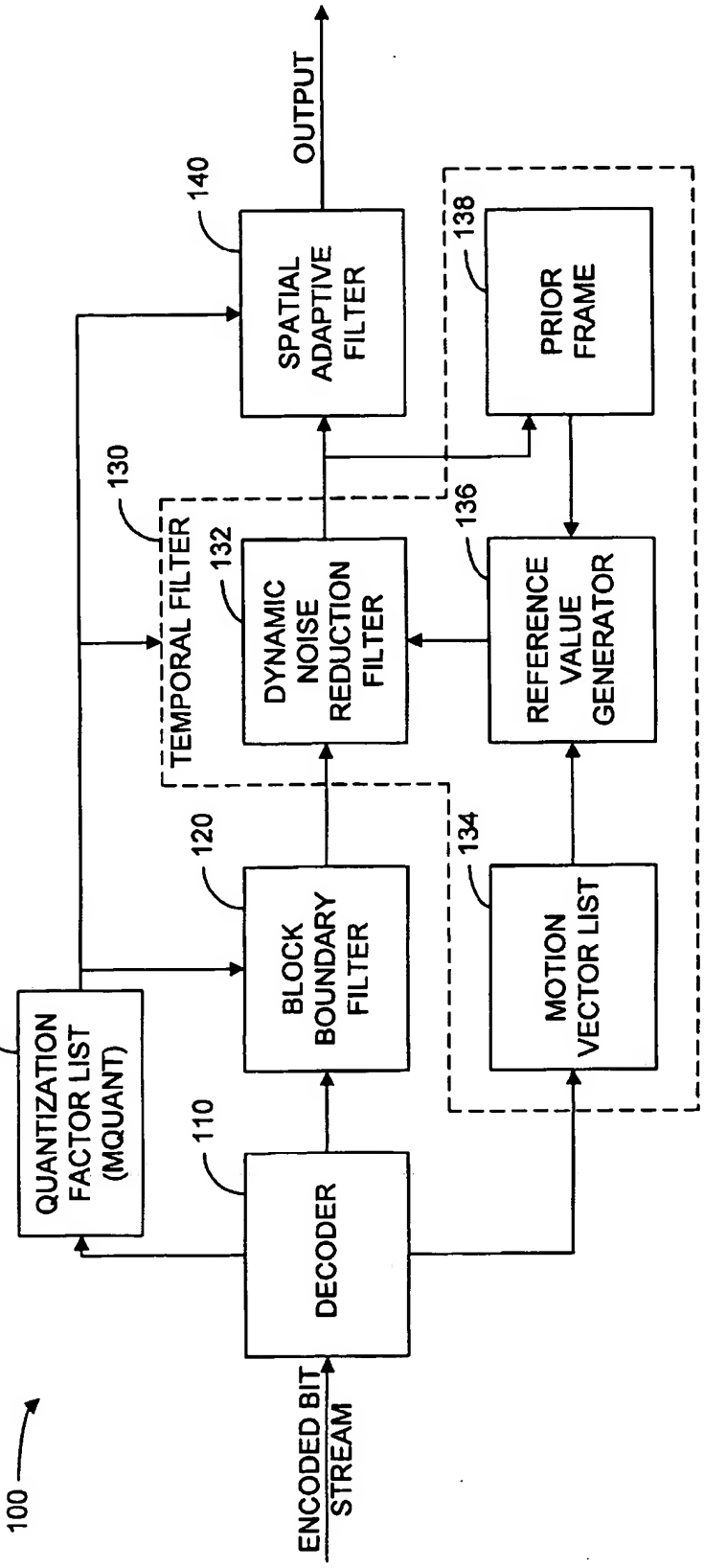

100
ENCODED BIT STREAM
DECODER
110
QUANTIZATION FACTOR LIST (MQUANT)
115
BLOCK BOUNDARY FILTER
120
MOTION VECTOR LIST
134
TEMPORAL FILTER
130
DYNAMIC NOISE REDUCTION FILTER
132
REFERENCE VALUE GENERATOR
136
PRIOR FRAME
138
SPATIAL ADAPTIVE FILTER
140
OUTPUT

FIG. 1

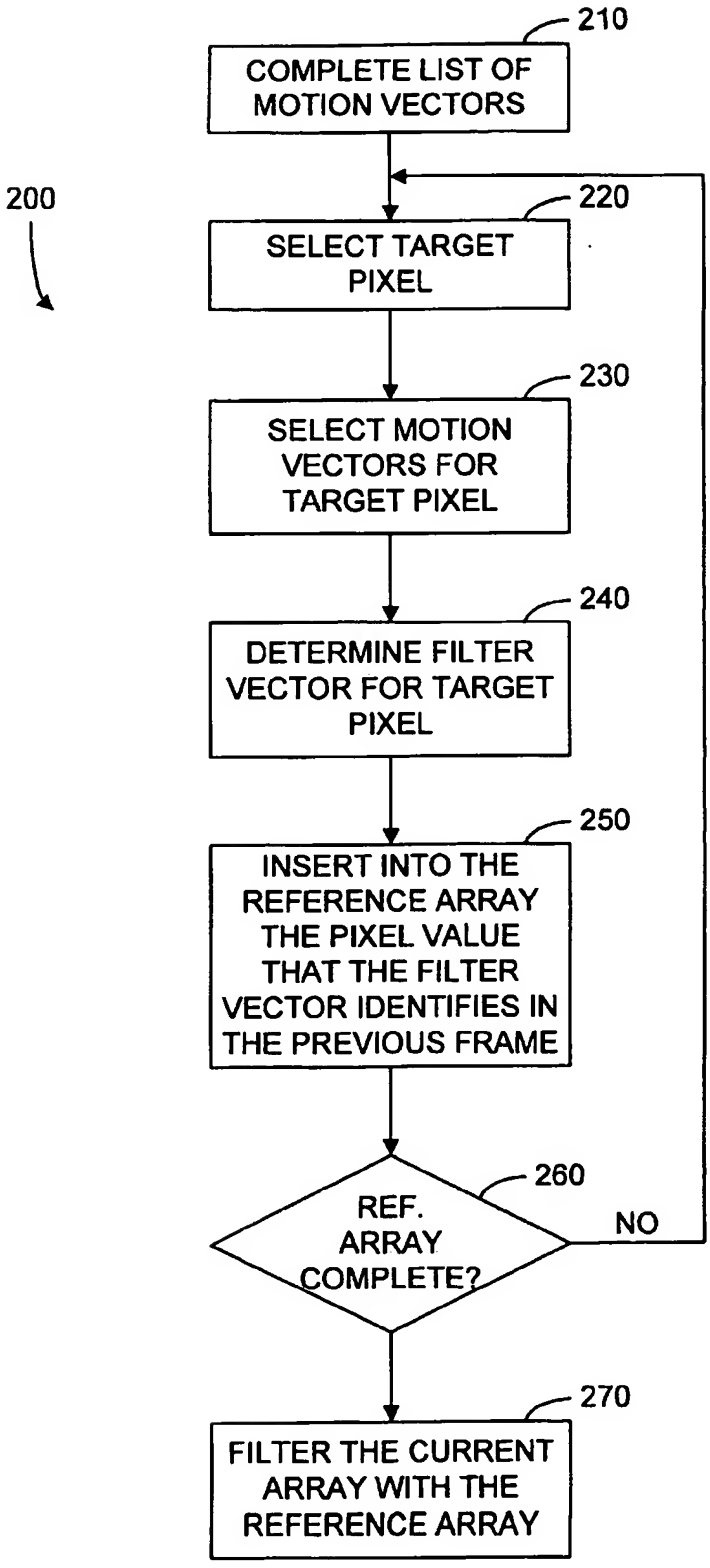

200
210
COMPLETE LIST OF MOTION VECTORS
220
SELECT TARGET PIXEL
230
SELECT MOTION VECTORS FOR TARGET PIXEL
240
DETERMINE FILTER VECTOR FOR TARGET PIXEL
250
INSERT INTO THE REFERENCE ARRAY THE PIXEL VALUE THAT THE FILTER VECTOR IDENTIFIES IN THE PREVIOUS FRAME
260
REF. ARRAY COMPLETE?
NO
270
FILTER THE CURRENT ARRAY WITH THE REFERENCE ARRAY

FIG. 2

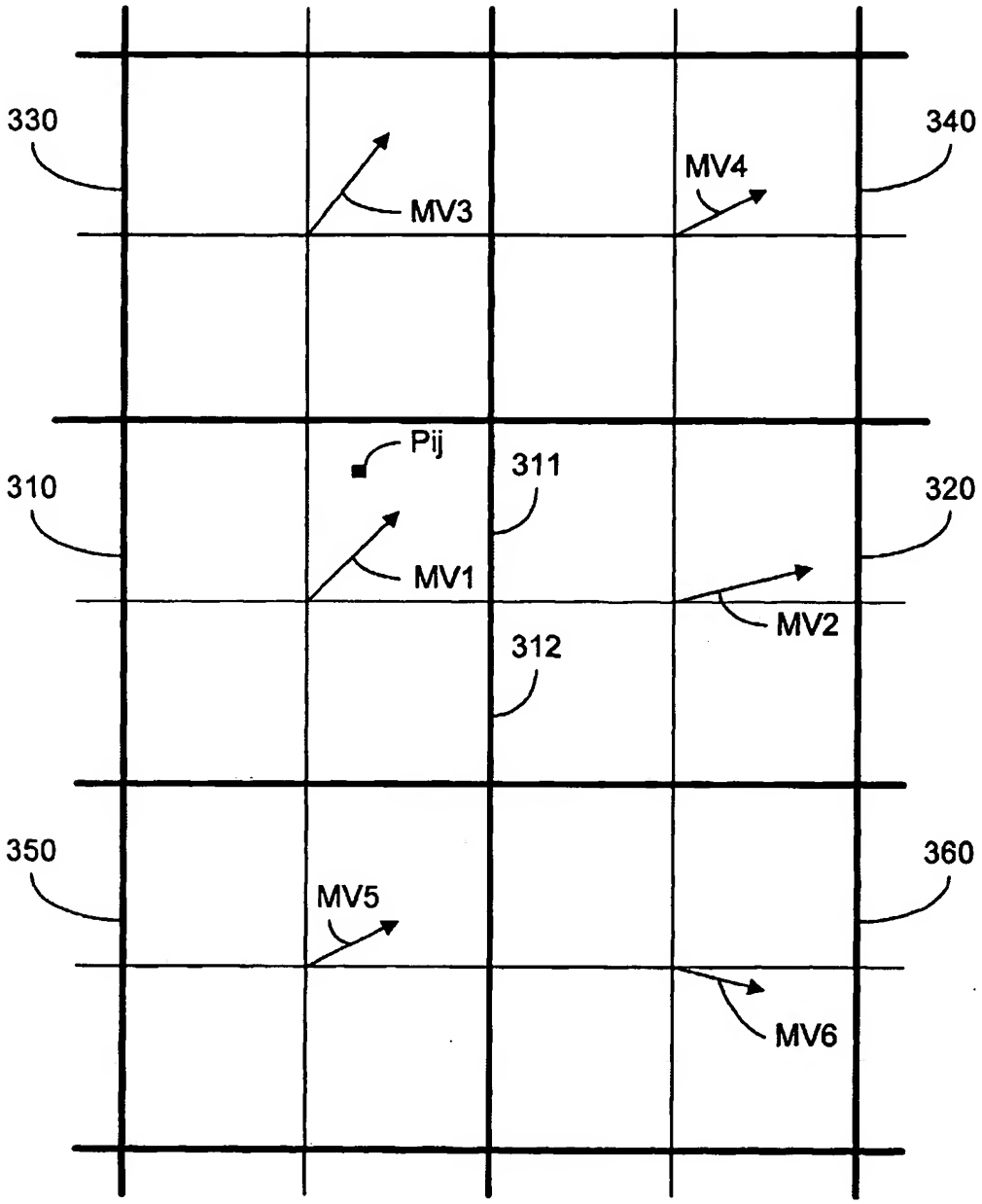


FIG. 3

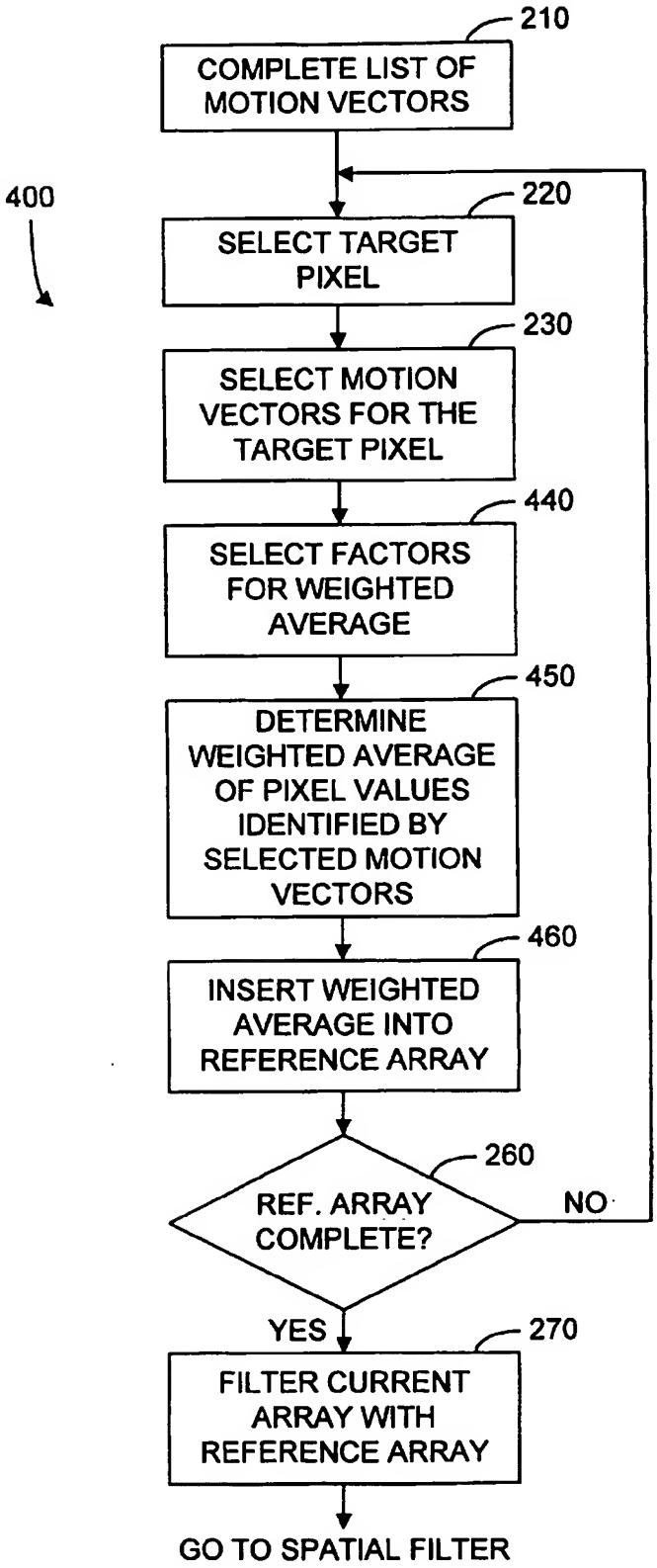
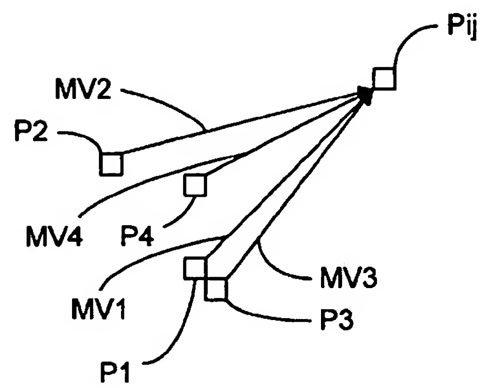
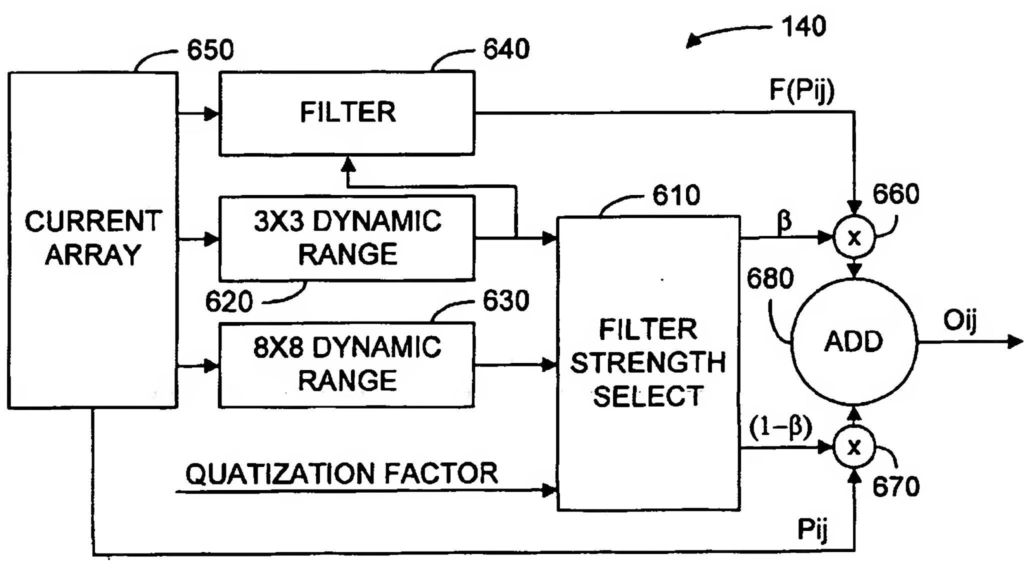

400
210
COMPLETE LIST OF MOTION VECTORS
220
SELECT TARGET PIXEL
230
SELECT MOTION VECTORS FOR THE TARGET PIXEL
440
SELECT FACTORS FOR WEIGHTED AVERAGE
450
DETERMINE WEIGHTED AVERAGE OF PIXEL VALUES IDENTIFIED BY SELECTED MOTION VECTORS
460
INSERT WEIGHTED AVERAGE INTO REFERENCE ARRAY
260
REF. ARRAY COMPLETE?
NO
YES
270
FILTER CURRENT ARRAY WITH REFERENCE ARRAY
GO TO SPATIAL FILTER

FIG. 4



Pij
MV2
P2
MV4
P4
MV1
MV3
P3
P1

FIG. 5



140
650
640
FILTER
F(Pij)
CURRENT ARRAY
3X3 DYNAMIC RANGE
610
β
660
620
630
680
8X8 DYNAMIC RANGE
FILTER STRENGTH SELECT
ADD
Oij
(1-β)
QUATIZATION FACTOR
670
Pij

FIG. 6

VIDEO POSTFILTERING WITH MOTION-COMPENSATED TEMPORAL FILTERING AND/OR SPATIAL-ADAPTIVE FILTERING

REFERENCE TO MICROFICHE APPENDIX

The present specification comprises a microfiche appendix. The total number of microfiche sheets in the microfiche appendix is one. The total number of frames in the microfiche appendix is 49.

COPYRIGHT NOTICE

BACKGROUND

1. Field of the Invention

This invention relates to systems for decoding video images and particularly to methods for improving decoded video image quality by removing coding artifacts and noise.

2. Description of Related Art

"Coding artifacts" are visible degradations in image quality that may appear as a result of encoding and then decoding a video image using a video compression method such as employed for the MPEG-1, MPEG-2, H.261, or H.263 standard. For example, video encoding for each of the MPEG-1, MPEG-2, H.261, and H.263 standards employs some combination of: partitioning frames of a video image into blocks; determining motion vectors for motion compensation of the blocks;

performing a frequency transform (e.g., a discrete cosine transform) on each block or motion-difference block; and quantizing the resultant transform coefficients. Upon decoding, common coding artifacts in a video image include blockiness that results from discontinuity of block-based motion compensation and inverse frequency transforms at block boundaries and "mosquito" noise surrounding objects in the video image as a result of quantization errors changing transform coefficients. Sources other than encoding and decoding can also introduce noise that degrades image quality. For example, transmission errors or noise in the system recording a video image can create random noise in the video image.

Postfiltering of a video image processes the video image to improve image quality by removing coding artifacts and noise. For example, spatial postfiltering can smooth the discontinuity at block boundaries and reduce the prominence of noise. Such spatial filtering operates on an array of pixel values representing a frame in the video image and modifies at least some pixel values based on neighboring pixel values. Spatial filtering can be applied uniformly or selectively to specific regions in a frame. For example, selective spatial filtering at a block edge (known locations within a frame) smoothes image contrast to reduce blockiness. However, spatial filtering can undesirably make edges and textures of objects in the image look fuzzy or indistinct and selective spatial filtering can cause "flashing" where the clarity of the edges of an object change as the object moves through areas filtered differently.

Temporal filtering operates on a current array of pixel values representing a current frame and combines pixel values from the current array with pixel values from one of more arrays representing prior or subsequent frames. Typically, temporal filtering combines a pixel value in the current array with pixel values in the same relative position in an array representing a prior frame under the assumption that the area remains visually similar. If noise or a coding artifact affects a pixel value in the current array but not the related pixel values in the prior frames, temporal encoding reduces the prominence of the noise or coding artifacts. A problem with temporal encoding arises from motion in the video image where the content of the image in one frame shifts in the next frame so that temporal filtering combines pixels in the current frame with visually dissimilar pixels in prior frames. When this occurs, the contribution of the dissimilar pixels creates a ghost of a prior frame in the current frame. Accordingly, temporal filtering can introduce undesired artifacts in a video image.

Postfiltering processes are sought that better remove coding artifacts and noise while preserving image features and not introducing further degradations.

SUMMARY

In accordance with the invention, a video postfiltering process includes motion compensated temporal filtering and/or spatial adaptive filtering. The motion compensated temporal filtering operates on each target pixel value in an array representing a current frame of a video image and combines each target pixel value with one or more pixel values from an array representing a prior frame. The pixel values from the prior frame alone or in combinations are sometimes referred to herein as reference values. The reference values for a target pixel in the current array are selected according to and depending on the values of a motion vector for a block containing the target pixel value and motion vectors for neighboring blocks. Using the motion vectors of neighboring blocks in the selection of reference values reduces ghosting when compared to temporal filtering without motion vectors or using only the motion vector for the block containing the target pixel.

In one embodiment of the invention, a vector (sometimes referred to herein as a filter vector) for a target pixel is determined from a weighted average of the motion vectors for the block containing the target pixel and the neighboring blocks closest to the target pixel. The weighting factors used in determining the filter vector for the target pixel depend on the position of the target pixel within a block. A pixel value for the target pixel is then filtered or combined with one or more reference values that correspond to an area of the prior frame identified by the filter vector.

An alternative embodiment of temporal filtering combines each target pixel value with pixel values from a prior frame that are in areas identified by the motion vectors for the block containing the target pixel value and neighboring blocks. The pixel values from the prior frame may be combined in a weighted average using weighting factors selected according to the position of the target pixel value within a block.

The adaptive spatial filter selects a filter operation for a target pixel according to the level of coding artifacts and the presence of important features. The level of coding artifacts depends on how well the pixel values are coded as indicated by the quantization factor. The dynamic range of the smallest coding unit, a 8×8 block in most of the standardized encoding processes, is used to estimate the amount of coding noise in the block. A large dynamic range usually indicates more noise. To reduce blurring of image features, a second

dynamic range around the target pixel is computed and used in two ways. The second dynamic range indicates the shape of the filter required to avoid mixing pixels from different features together. The second dynamic range also indicates the appropriate strength of the filter. When the second dynamic range is close to the first dynamic range, the target pixel is on or near image features, and a weak filter is used. When the second dynamic range is smaller than a large first dynamic range, the target pixel is likely to be noise around the edges and a strong filter is used. Other combinations of the sizes of the dynamic ranges result in the use of other filters.

Although the temporal filtering and spatial filtering are used in combination to provide the best image quality, either may be used alone in particular embodiments of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows a video decoder implementing postfiltering in accordance with an embodiment of the invention.

FIG. **2** shows a flow diagram for a motion compensated temporal filtering process in accordance with an embodiment of the invention.

FIG. **3** illustrates motion vectors for a portion of a frame that is divided into blocks.

FIG. **4** shows a flow diagram for a motion compensated temporal filtering process in accordance with another embodiment of the invention.

FIG. **5** illustrates pixel values from a prior frame that are combined to form a reference value for a target pixel in a current frame.

FIG. **6** shows a spatial adaptive filter in accordance with an embodiment of the invention.

Use of the same reference symbols in different figures indicates similar or identical items.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In accordance with an aspect of the invention, a video postfilter employs motion compensated temporal filtering and spatial adaptive filtering to improve image quality and remove coding artifacts. The temporal filtering uses motion vectors from multiple blocks to determine a reference value that is combined with the target pixel value being filtered. The reference value selected using multiple motion vectors better matches the target pixel value because the combination of motion vectors can better approximate motion of individual pixels than can a motion vector that indicates average motion of an entire block of pixel values. The spatial adaptive filtering uses the dynamic ranges of pixel values in blocks of different sizes to determine the visual context of the target pixel, and selects a filter for the target pixel according to the determined visual context. Such postfiltering processes improve video image quality and are applicable to any video image. However, the postfiltering processes are particularly suited for postfiltering a video image decoded in accordance with a video standard such as the well-known MPEG-1, MPEG-2, H.261, or H.263 video standard.

FIG. 1 shows a block diagram of a decoding system 100 in accordance with an exemplary embodiment of the invention. Decoding system 100 may be implemented in software executed by a general purpose computer or in specialized hardware designed to implement the specific functions of system **100** as described herein. As a specific example of an application of the invention, decoding system **100** decodes a video signal complying with the H.261 standard for video conferencing. Alternative applications will be apparent in view of this disclosure.

In accordance with the H.261 standard, three arrays of pixel values, a Y array, a U array, and a V array, represent each frame of the video image and are respectively associated with luma Y and chroma U and V of associated pixels in the frame. In the exemplary embodiment of the invention shown in FIG. **1**, only the Y arrays of frames are postfiltered since the Y arrays have the greatest influence on the appearance of the video image. The Y array for a frame contains **288** rows and 352 columns of pixel values where each pixel value indicates the luma for a pixel in a standard frame size, which is 288×352 pixels for the H.261 CIF picture format. The U and V arrays each contain 144 rows and 176 columns of pixel values where each pixel value indicates U or V chroma for four pixels in the associated frame. The H.261 standard partitions each frame into 16×16-pixel areas where each area is represented by a macroblock of pixel values. Each macroblock includes one 16×16 block (or four 8×8 blocks) from the Y array, one 8×8 block from the U array, and one 8×8 block from the V array. During encoding of each macroblock in a current frame, an encoder uses a match criterion such as mean square error, mean quadratic error, or mean absolute error to search for a 16×16-pixel area in a prior frame that is visually similar to the 16×16-pixel area associated with the macroblock. A motion vector for the macroblock indicates an offset from the position of the similar area in the prior frame to the position of the associated area in the current frame. Each macroblock is then intercoded or intracoded depending on whether the search finds a good match (i.e., similar) area in the prior frame. Intercoding determines a difference block that is the difference between a block representing the area in the current frame and a block in the prior frame indicated by the motion vector, breaks the difference block into 8×8 difference blocks, and performs a discrete cosine transform (DCT) on each of the 8×8 difference blocks. Intracoding performs a discrete cosine transform on the 8×8 blocks of pixel values rather than on the difference blocks. Following intercoding or intracoding, the transform coefficients are quantized and then transmitted with motion vectors for the intercoded blocks in an encoded bit stream representing the video image.

Decoding system **100** includes a decoder **110**, a block boundary filter (BBF) **120**, a motion compensated temporal filter **130**, and a spatial adaptive filter **140** that process an incoming bit stream complying with the H.261 standard. Decoder **110** is a conventional decoder that decodes the bit stream to generate arrays of pixel values representing decoded frames that form a video image. In the decoding, decoder **110** identifies a quantization factor MQUANT from a quantization factor list **115**, dequantizes transform coefficients, performs an inverse discrete cosine transformation (IDCT) on 8×8 blocks of dequantized transformation coefficients, and for intercoded blocks, sums the resulting difference blocks with the similar blocks that the motion vectors identify in the prior frame. BBF **120**, motion compensated temporal filter **130**, and spatial adaptive filter **140** postfilter the decoded video image from decoder **110** to reduce noise and coding artifacts and improve image quality.

Block boundary filter **120** reduces blockiness resulting from discontinuity at the boundaries of 8×8 blocks that were subject to independent DCTs and changes the pixel values at the boundaries of the 8×8 blocks to smooth transitions across the block boundaries. Specifically, if the columns of

an array of pixel values are numbered from 0 to 351, BBF **120** changes pixel values in columns 8n and 8n+7 for $0 \leqq n \leqq 43$; and if the rows are numbered from 0 to 287, BBF **120** changes pixel values in rows 8m and 8m+7 for $0 \leqq m \leqq 35$. In the exemplary embodiment of the invention, BBF **120** adapts or changes according to the quantization step size MQUANT for the block containing the pixel being filtered. For the H.261 standard, the encoded bit stream indicates the quantization step size for each encoded block. Table 1 indicates the coefficients for a five-tap horizontal filter for use on a target pixel in a column j and a three-tap vertical filter for use on a target pixel in a row i.

TABLE 1

	Block Boundary Filter	
MQUANT	Horz. Filter Taps j-2, j-1, j, j + 1, j + 2	Vertical Filter Taps i-1, i, i + 1
1	0, 0, 32, 0, 0	0, 32, 0
2	1, 2, 26, 2, 1	1, 30, 1
3	1, 4, 22, 4, 1	2, 28, 2
4	2, 4, 20, 4, 2	3, 26, 3
5	2, 5, 18, 5, 2	4, 24, 4
6	2, 6, 16, 6, 2	5, 22, 5
7	2, 7, 14, 7, 2	6, 20, 6
>7	2, 8, 12, 8, 2	7, 18, 7

The filter coefficients for each filter in Table 1 sum to 32 so that the result of a filter operation on a target pixel value Pij is the sum of products of pixel values and filter coefficients right shifted by 5-bits (i.e., divided by 32.)

BBF **120** provides pixel values of a block-boundary-filtered frame to motion compensated temporal filter **130**. Motion compensated temporal filter **130** includes a dynamic noise reduction filter **132** that combines pixel values for the current frame with reference values derived from pixel values for a prior frame **138**. A reference value generator **136** determines the reference value using a list **134** of motion vectors for the current frame and the pixel values for prior frame **138**.

FIG. **2** illustrates a flow diagram of a temporal filtering process **200** that filter **130** implements. An initial step **210** in process **200** completes the motion vector list **134** for the current frame. In the exemplary embodiment, decoder **110** write to list **134** the motion vectors as decoded from the incoming bit stream. Alternatively, temporal filter **130** can determine the motion vectors from the pixel values for the decoded current and prior frames, but determining motion vectors increases filter complexity. Additionally, the decoded motion vectors from decoder **110** typically provide better temporal filtering because an encoder selected encoded motion vectors using image data before compression. When the current frame is fully decoded, each macroblock not having a decoded motion vector (e.g., each intracoded macroblock) is assigned a motion vector of length zero or an illegal motion vector. Subsequent steps in process **200** skip intracoded microblocks and replace each illegal motion vector with a motion vector for a neighboring block.

Once list **134** is complete, reference value generator **136** in step **220** selects a target pixel in the current frame and identifies a macroblock containing the target pixel. A step **230** selects motion vectors for the target pixel, and a step **240** uses the selected motion vectors to determine a filter vector for the target pixel. The filter vector indicates an offset to the position of the target pixel from the position of a pixel in the prior frame that will be combined with the target pixel in a filter operation. In accordance with an aspect of the invention, the filter vector is derived using a weighted average of the motion vectors for the macroblock containing the target pixel and neighboring macroblocks. For example, Equation 1 defines a filter vector FVij for target pixel Pij in the exemplary embodiment of the invention.

$$FVij=\text{round}\,(Aij*MVA+Bij*MVB+Cij*MVC+Dij*MVD) \quad \text{Equation 1}$$

In Equation 1, Aij, Bij, Cij, and Dij are weighting factors, MVA, MVB, MVC, and MVD are the motion vectors selected in step **230**, and round( ) is a function that rounds its argument to the nearest integer.

To illustrate the selection of motion vectors, FIG. **3** shows a portion of a current frame represented by six 16x16-pixel areas **310**, **320**, **330**, **340**, **350**, and **360**. Macroblocks representing areas **310**, **320**, **330**, **340**, **350**, and **360** have respective motion vectors MV1, MV2, MV3, MV4, MV5, and MV6 that identify visually similar 16x16-pixel areas in the prior frame. For a specific target pixel, the motion vector for the macroblock representing the target pixel may not indicate a similar pixel in the prior frame if motion of an object including target pixel differs from the average motion for the block. The motion of the target pixel may be more like the average motion of a neighboring block rather than the block containing the target pixel. Accordingly, in the exemplary embodiment of the invention, motion vectors MVA, MVB, MVC, and MVD are respectively the motion vector for the block containing the target pixel, the motion vector for the nearest neighboring block to the right or left of the target pixel, the motion vector for the nearest neighboring block above or below the target pixel, and the motion vector for the nearest neighboring block on a diagonal relative to the block containing of the target pixel. For example, when target pixel value Pij is in an upper-right quadrant **311** of block **310** as illustrated in FIG. **3**, selected motion vectors MVA, MVB, MVC, and MVD of Equation 1 are respectively motion vectors MV1, MV2, MV3, and MV4. If target pixel value Pij were in lower-right quadrant **312**, the selected motion vectors MVA, MVB, MVC, and MVD would respectively be motion vectors MV1, MV2, MV5, and MV6. If a motion vector MVB, MVC, or MVD would otherwise correspond to a block beyond the edge of the frame, a block in the frame but closest to the desired block provides that motion vector.

In Equation 1, weighting factors Aij, Bij, Cij, and Dij depend on indices i and j which respectively indicate the vertical and horizontal positions of the target pixel Pij in a quadrant of a block. Indices i and j range from 1 to 8 for an 8x8 quadrant containing target pixel Pij and have minimum values near the center of 16x16 block. Equations 2 give the self contribution weighing factor Aij, the right/left neighbor weighting factor Bij, the upper/lower neighbor weighting factor Cij, and the diagonal neighbor weighting factor Dij for an exemplary embodiment of the invention.

$$Aij=(16.5-i)*(16.5-j)/256$$

$$Bij=(16.5-i)*(j-0.5)/256$$

$$Cij=(i-0.5)*(16.5-j)/256$$

$$Dij=(i-0.5)*(j-0.5)/256 \quad \text{Equations 2}$$

The weighting factors for the possible target pixel locations in an 8x8 quadrant are selected according to the likelihood that the motion of an object including target pixel Pij is similar to the motion vector associated with the weighting factor. For example, if target vector Pij is near the center of block **310**, the motion of target pixel Pij is likely to be

similar to motion vector MV1. Accordingly, weighting factor Aij dominates the other weighting factors when indices i and j indicate a target point near the center of block **310** (i.e., if i and j are both at or near 1.) As index j or i increases, target pixel Pij nears the boundary of block **320** or **330**, and coefficient Bij or Cij increases the contribution of motion vector MV2 or MV3.

In process **200** (FIG. **2**), a step **250** uses the filter vector to identify a reference value that is inserted into a reference array. The inserted value is inserted at the position corresponding to the target pixel but is from a position offset from the position of the target pixel by the amount indicated by the filter vector. Steps **220** to **250** are repeated for each pixel in the current frame until the reference array is complete in step **260**. A filtering step **270** combines pixel values from the array representing the current decoded frame with reference values from the reference array. Equation 3 indicates the form of a filtering that combines target pixel value Pij for the current frame and a reference value Rij from the reference array to generate an output pixel value Oij.

$$Oij=Pij-F(Pij-Rij) \qquad \text{Equation 3}$$

Filter function F(Pij−Rij) is a function of a difference Δ between decoded pixel value Pij and the associated reference value Rij. For a large difference A, filter function F(Δ) is zero so that no temporal filtering is performed if reference value Rij is not a good match for decoded pixel value Pij. The filter function F(Δ) may further depend on coding parameters such as the macroblock quantization step size Q. Table 2 illustrates a filter function F(Δ,Q) suitable for the exemplary embodiment of the invention.

TABLE 2

Filter Function F (Δ,Q) of Difference Δ and Quantization Step Q

Δ\Q	1	2	3	4	5	6	7	8	9	>9
0	0	0	0	0	0	0	0	0	0	0
1	0.29	0.34	0.39	0.44	0.49	0.54	0.59	0.64	0.69	0.74
2	0.44	0.52	0.59	0.67	0.74	0.81	0.89	0.96	1.04	1.11
3	0.87	1.01	1.16	1.30	1.45	1.59	1.74	1.88	2.03	2.17
4	1.16	1.36	1.55	1.75	1.94	2.14	2.33	2.53	2.72	2.92
5	1.34	1.57	1.79	2.01	2.24	2.46	2.69	2.91	3.14	3.36
6	1.53	1.79	2.04	2.30	2.56	2.81	3.07	3.33	3.58	3.84
7	1.73	2.02	2.30	2.59	2.88	3.17	3.46	3.75	4.04	4.32
8	1.93	2.25	2.57	2.90	3.22	3.54	3.86	4.19	4.51	4.83
9	2.11	2.47	2.82	3.17	3.52	3.88	4.23	4.58	4.94	5.29
10	2.32	2.71	3.10	3.49	3.88	4.27	4.65	5.04	5.43	5.82
11	2.08	2.43	2.78	3.13	3.48	3.83	4.17	4.52	4.87	5.22
12	1.89	2.21	2.52	2.84	3.16	3.47	3.79	4.10	4.42	4.74
13	1.69	1.97	2.25	2.53	2.81	3.10	3.38	3.66	3.94	4.22
14	1.48	1.72	1.97	2.22	2.47	2.71	2.96	3.21	3.45	3.70
15	1.30	1.52	1.74	1.95	2.17	2.39	2.61	2.82	3.04	3.26
16	1.11	1.30	1.49	1.67	1.86	2.05	2.23	2.42	2.61	2.80
17	0.93	1.09	1.24	1.40	1.55	1.71	1.86	2.02	2.18	2.33
18	0.74	0.87	0.99	1.12	1.24	1.37	1.49	1.62	1.74	1.87
19	0.56	0.65	0.75	0.84	0.94	1.03	1.12	1.22	1.31	1.41
20	0.37	0.44	0.50	0.56	0.63	0.69	0.75	0.82	0.88	0.94
21	0.19	0.22	0.25	0.29	0.32	0.35	0.38	0.42	0.45	0.48
>21	0	0	0	0	0	0	0	0	0	0

The filter function coefficients and reference values can be stored using double precision, e.g. 16-bits of precision where 8 bits are normally used for pixel values to reduce rounding errors.

The exemplary embodiment of the temporal filtering process illustrated in FIG. **2** and described above may be varied in a number of ways in keeping with the invention. For example, step **230** may select more or fewer that four motion vectors per target pixel. In particular, step **230** could select three motion vectors (the motion vectors for the block containing the target pixel, the nearest neighboring block to the left or right of the target pixel, and the nearest neighboring block above or below the target pixel) or nine motion vectors (the motion vectors for the block containing the target pixel, the eight nearest neighboring blocks.) Further, determining the filter vector in step **240** can use a variety of different weighting factors or functions of the selected motion vectors and is not limited to a weighted average or particular weighting factors. Additionally, each reference value can be combined with a target pixel in a filtering operation immediately after step **240** without ever generating the reference array. Further, a variety of filter functions not limited to the form of Equation 3 described above may be employed. For example, filters can combine each target pixel with more that one reference value from the reference array.

FIG. **4** illustrates an alternative temporal filtering process **400** in accordance with the invention. Process **400** begins with the same steps **210**, **220**, and **230** described above in reference to FIG. **2**. Step **210** completes the list **134** of motion vectors for macroblocks representing the current frame. Step **220** selects a target pixels in the current decoded frame, and step **230** selects a set of motion vectors from list **134** for the target pixel. The selected motion vectors include, for example, the motion vector for the block containing the target pixel, the motion vector for the nearest neighboring block to the left or right of the target pixel, the motion vector for the nearest neighboring block above or below the target pixel, and the motion vector for the nearest neighboring block at a diagonal with the block containing the target pixel. For example, referring to target pixel Pij in FIG. **3**, step **230** selects motion vectors MV1, MV2, MV3, and MV4.

With one end at the target pixel, each of the selected motion vectors identifies a pixel value in the array representing the prior frame. FIG. **5** shows four pixels P1, P2, P3, and P4 which respective motion vectors MV1, MV2, MV3, and MV4 for target pixel Pij. The pixel values that the selected motion vectors identify in the prior frame are combined with the target pixel value in a filter operation. For process **400**, the filter operation combines the target pixel value with a reference value that is a weighted average of the

pixel values that the selected motion vectors identify. Step **440** selects the factors for the weighted average, and step **450** determines the weighted average that will be a reference value. For example, Equation 4 defines a reference value Rij for a target pixel Pij.

$$Rij=Aij*PA+Bij*PB+Cij*PC+Dij*PD \quad \text{Equation 4}$$

In Equation 4, Aij, Bij, Cij, and Dij are the factors for the weighted average and may be, for example, as defined in Equation 2 above. PA, PB, PC, and PD are pixel values for the prior frame that motion vectors MVA, MVB, MBC, and MVD identify for target pixel value Pij. Motion vectors MVA, MVB, MBC, and MVD are the motion vectors respectively for the block containing the target pixel value, the nearest neighboring block to the left or right of the target pixel, the nearest neighboring block above or below the target pixel, and the nearest neighboring block at a diagonal with the block containing the target pixel. For target pixel Pij in quadrant **311** as illustrated in FIG. **3**, the selected motion vectors MVA, MVB, MBC, and MVD are respectively motion vectors MV1, MV2, MBA, and MV4, and pixel values PA, PB, PC, and PD are pixel values P1, P2, P3, and P4 in the prior frame at positions illustrated in FIG. **5**. Step **460** inserts the reference value Rij into the reference array. When the reference array is complete, step **270** combines current with the reference array in a filter operation such as defined by Equation 3 and Table 2 above. Alternatively, each pixel value Pij can be combined with reference values Rij as the reference values become available.

After filtering every pixel in the current array, temporal filter **130** rounds the filtered current array to normal pixel value precision (e.g., 8-bits) and provides the rounded array to spatial adaptive filter **140**. FIG. **6** illustrates an embodiment of spatial adaptive filter **140**. Spatial adaptive filter **140** includes a filter strength select unit **610** that selects a filter strength for the filtering of each target pixel Pij from a current frame **650**. Filter strength select unit **610** bases selection of the filter strength on a dynamic range DR3 of pixel values in a smaller block containing the target pixel, a dynamic range DR8 of pixel values in a larger block containing the target pixel, and the quantization step size MQUANT for the macroblock containing the target pixel. A dynamic range is the difference between the largest and the smallest pixel values in an area. In the embodiment of FIG. **6**, the smaller block is a 3×3 block centered on the target pixel value, and the larger block is an 8×8 block that was subjected to a DCT during encoding. It has been found that similarities and differences between dynamic ranges DR3 and DR8 for the smaller and larger blocks suggest the image content of the area including and surrounding the target pixel. Filter select unit **610** selects a filter as appropriate for the image content suggested by the dynamic ranges. For example, a large dynamic range suggests that the associated block contains an edge of an object in the frame. The smaller block having a relatively small dynamic range DR3 and the larger block having a relatively large dynamic range DR8 suggests that the larger block contains an edge of an object and the smaller block is near but does not contain a portion of that edge. In this case, the target pixel is strongly filtered because coding artifacts are common near sharp edges within a block that has been DCT transformed. A 3×3 region containing a large dynamic range suggests that the target pixel is at the edge of an object. In this case the target pixel is weakly filtered to avoid blurring of the edge. Both dynamic ranges DR3 and DR8 being moderate suggests that the target pixel is part of texture in the image frame, and a weak filter is applied to the target pixel to avoid blurring the texture. Table 3 shows combinations of the dynamic ranges, the image content suggested by each combination, and the appropriate level of filtering for each combination.

TABLE 3

	Filter Selection		
	DR8 is small	DR8 is moderate	DR8 is large
DR3 is small	Weak Filter: Target could be noise or detail	Medium Filter: Target likely noise on detail	Strong Filter: Target likely noise near an edge
DR3 is moderate	Very Weak Filter:	Weak Filter: Target likely image texture	Medium filter: Target could be noise or detail
DR3 is large	Very Weak Filter:	Very Weak Filter:	Weak Filter: Target likely at an edge

The largest change between adjacent pixel values similarly measures image content, but determining the largest change is more complex than determining the dynamic range. To determine a dynamic range, units **620** and **630** determine the difference between the largest and smallest pixel values in respective 3×3 and 8×8 blocks. In the exemplary embodiment, each pixel value is an 8-bit value indicating the luma for a pixel so that the possible dynamic ranges are from 0 to 255. The dynamic range for the small block can be greater than the dynamic range for the larger block if the smaller block contains pixel values from outside the larger block.

To select the filter strength applied to a target pixel in the current frame, filter strength select unit **640** generates a parameter β, and the filter applied to a target pixel is of the form given in Equation 5.

$$Oij=\text{round\_and\_clip}((1-\beta)*Pij+\beta*F(Pij)) \quad \text{Equation 5}$$

In Equation 5, Oij is the output pixel value from filter **140** for target pixel Pij, F(Pij) is the output pixel value of a spatial filter **640** in filter **140**, and round_and_clip is a function that rounds its argument to the nearest integer and clips that result according to the range of allowed pixel value. Parameter β is restricted to a range from 0 to 1, where the strength of the filter increases with parameter β. For β equal to zero, output pixel value Oij is equal to target pixel Pij unfiltered. For β equal to one, output pixel value Oij is equal to the result F(Pij) from spatial filter **640**.

Filter **640** can be any desired spatial filter. In an exemplary embodiment of the invention, spatial filter is a "5×5 like filter" that excludes from the filter operation pixel values that significantly differ from a target pixel value being filtered. Table 4 illustrates the filter coefficients for the exemplary embodiment of filter **640**.

TABLE 4

	Filter Coefficients *32				
	j-2	j-1	j	j+1	j+2
i-2	0	1	1	1	0
i-1	1	2	2	2	1
i	2	2	2	2	2
i+1	1	2	2	2	1
i+2	0	1	1	1	0

F(Pij) is the sum of the product of the filter coefficients from Table 3 and pixel values. Each pixel value in a product is either the pixel value having a position relative to the target pixel as indicated for the filter coefficient in the product or

the target pixel value if the pixel value in the position indicated for the filter coefficient differs from the target pixel value by more than a likeness threshold LT. For the exemplary embodiment, Equation 6 shows the dependence of likeness threshold LT on the dynamic range DR3 of the 3×3 block in the exemplary embodiment of the invention.

$$LT=10+0.625*DR3 \qquad \text{Equation 6}$$

Tables 5.1 and 5.2 below indicate the selection of parameter β for different values of dynamic ranges DR3 and DR8 and the macroblock quantization step size MQUANT. Table 5.1 indicates the values of parameter β when the quantization step size is six. For quantization step sizes MQUANT less than six the values in Table 5.1 are scaled by MQUANT/6.

TABLE 5.1

Parameter β for MQUANT = 6

DR8 / DR3	<5	<10	<15	<20	<25	<30	<40	<50	<60	<70	<90	<120	<160	<256
<5	.25	.25	.25	.25	.25	.25	.32	.38	.44	1	1	1	1	1
<10	.10	.25	.25	.30	.30	.30	.35	.40	.45	1	1	1	1	1
<15	.05	.15	.30	.30	.30	.25	.30	.30	.35	.9	1	1	1	1
<20	0	0	.15	.15	.15	.10	.15	.20	.25	.6	.8	1	1	1
<25	0	0	0	0	0	0	.10	.15	.20	.5	.7	.9	1	1
<30	0	0	0	0	0	0	.05	.10	.15	.4	.5	.8	.9	1
<40	0	0	0	0	0	0	0	.05	.1	.3	.5	.7	.9	.9
<50	0	0	0	0	0	0	0	0	0	.3	.4	.5	.7	.8
<60	0	0	0	0	0	0	0	0	0	.3	.3	.4	.5	.6
<70	0	0	0	0	0	0	0	0	0	.3	.3	.3	.4	.4
<90	0	0	0	0	0	0	0	0	0	.3	.3	.3	.3	.3
<120	0	0	0	0	0	0	0	0	0	.3	.3	.3	.3	.3
<160	0	0	0	0	0	0	0	0	.3	.3	.3	.3	.3	0
<256	0	0	0	0	0	0	0	0	.3	.3	.3	.3	.3	0

Table 5.2 indicates the values of parameter β, for average quantization greater than 10.

TABLE 5.2

Parameter β for MQUANT ≥11

DR8 / DR3	<5	<10	<15	<20	<25	<30	<40	<50	<60	<70	<90	<120	<160	<256
<5	.5	.5	.5	.5	.5	.5	.63	.76	.89	1	1	1	1	1
<10	.2	.5	.5	.6	.6	.6	.7	.8	.9	1	1	1	1	1
<15	.1	.3	.6	.6	.6	.5	.6	.6	.7	.9	1	1	1	1
<20	.1	.1	.3	.3	.3	.2	.3	.4	.5	.6	.8	1	1	1
<25	.1	.1	.1	.1	.1	.1	.2	.3	.4	.5	.7	.9	1	1
<30	.1	.1	.1	.1	.1	.1	.1	.2	.3	.4	.5	.8	.9	1
<40	.1	.1	.1	.1	.1	.1	.1	.1	.2	.3	.5	.7	.9	.9
<50	.1	.1	.1	.1	.1	.1	.1	.1	.1	.3	.4	.5	.7	.8
<60	.1	.1	.1	.1	.1	.1	.1	.1	.1	.3	.3	.4	.5	.6
<70	.1	.1	.1	.1	.1	.1	.1	.1	.1	.3	.3	.3	.4	.4
<90	.1	.1	.1	.1	.1	.1	.1	.1	.1	.3	.3	.3	.3	.3
<120	.1	.1	.1	.1	.1	.1	.1	.1	.1	.3	.3	.3	.3	.3
<160	.1	.1	.1	..1	.1	.1	.1	.1	.1	.3	.3	.3	.3	.3
<256	.1	.1	.1	.1	.1	.1	.1	.1	.1	.3	.3	.3	.3	.3

For quantization step size MQUANT greater than 6 but less than 11, parameter β is determined by linear interpolation between a value from Table 5.1 and a value from Table 5.2.

The microfiche appendix contains a C-language program listing for a software embodiment of a postfilter in accordance with an exemplary embodiment of the invention.

Although the invention has been described with reference to particular embodiments, the description is only an example of the invention's application and should not be taken as a limitation. Various adaptations and combinations of features of the embodiments disclosed are within the scope of the invention as defined by the following claims.

We claim:

1. A method for improving appearance of a video image, comprising:

representing a first frame in the video image by a first array of pixel values and a second frame in the video image by a second array of pixel values;

selecting a plurality of motion vectors for a target pixel value in the first array, wherein each motion vector corresponds to a block of pixel values in the first array and identifies a block of pixel values in the second array;

determining a reference value for the target pixel value, wherein the reference value depends on the motion vectors selected for the target pixel value and one or more pixel values from the second array; and

combining the target pixel value with the reference value in a filter operation that generates an output pixel value for a third array, the third array representing improved version of the first frame, wherein the output pixel value is equal to the target pixel value if a difference between the target value and the reference value is

greater than a threshold value and is equal to a linear combination of the target pixel value and the reference value if the difference is not greater than the threshold value.

2. The method of claim 1, wherein selecting the motion vectors comprises:

selecting a first motion vector that corresponds to a first block containing the target pixel value; and

selecting a second motion vector that corresponds to a second block neighboring the first block.

3. The method of claim 2, wherein the second block abuts the first block, and of blocks that abut the first block, the second block has a boundary closest to the target pixel value.

4. The method of claim 3, wherein determining the reference value for the target pixel value comprises:

combining the motion vectors selected for the target pixel value to generate a filter vector; and

selecting as the reference value a pixel value in the second array, at a position offset from a position of the target pixel value by an amount indicated by the filter vector.

5. The method of claim 4, wherein combining the motion vectors comprises:

selecting weighting factors that depend on the position of the target pixel value in the first array; and

determining a weighted average of the motion vectors using the selected weighting factors.

6. The method of claim 3, wherein determining the reference value for the target pixel value comprises:

for each of the motion vectors selected for the target pixel value, identifying a pixel value that is in the second array, at a position that is offset from a position corresponding to the target pixel value by an amount indicated by the motion vector; and

combining the pixel values identified to determine the reference value for the target pixel value.

7. The method of claim 6, wherein combining the pixel values comprises:

selecting weighting factors that depend on the position of the target pixel value in the first array; and

determining a weighted average of the identified pixel values using the selected weighting factors.

8. The method of claim 1, wherein determining the reference value for the target pixel value comprises:

combining the motion vectors selected for the target pixel value to generate a filter vector; and

selecting as the reference value a pixel value from the second array, wherein the pixel value selected is in the second array, at a position that is offset from a position corresponding to the target pixel value by an amount indicated by the filter vector.

9. The method of claim 8, wherein combining the motion vectors comprises:

selecting weighting factors that depend on the position of the target pixel value in the first array; and

determining a weighted average of the motion vectors using the selected weighting factors.

10. The method of claim 1, wherein determining the reference value for the target pixel value comprises:

for each of the motion vectors selected for the target pixel value, identifying a pixel value that is in the second array, at a position offset from a position corresponding to the target pixel value by an amount indicated by the motion vector; and

combining the pixel values identified to determine the reference value for the target pixel value.

11. The method of claim 10, wherein combining the pixel values comprises:

selecting weighting factors that depend on the position of the target pixel value in the first array; and

determining a weighted average of the identified pixel values using the selected weighting factors.

12. The method of claim 1, further comprising decoding a bit stream representing the video image, wherein:

the decoding exacts from the bit stream motion vectors that are required for further decoding of the bit stream; and

selecting the plurality of motion vectors for the target pixel value comprises selecting a motion vector extracted from the bit stream.

13. The method of claim 12, wherein the bit stream is encoded according to a video standard selected from a group consisting of the MPEG-1 standard, the MPEG-2 standard, the H.261 standard, and the H.263 standard.

14. The method of claim 12, wherein decoding further includes determining a quantization factor from the bit stream.

15. The method of claim 1, further comprising for each pixel value in the first array, repeating the selecting, determining, and combining steps with the pixel value as the target pixel value.

16. A method for improving appearance of a video image, comprising:

determining motion vectors for first areas in a first frame of the video image that is represented by a first array of pixel values, each motion vector corresponding to a first area in the first frame and a second area in a second frame, wherein image content of the second area in the second frame is similar to the image content of the first area in the first frame;

determining for each pixel in the first frame a reference vector that is a combination of a motion vector for a first area containing the pixel and one or more of the motion vectors for adjacent first areas;

generating a reference array containing reference values, wherein each reference value in the reference array is equal to the pixel value at a relative position in the second array that is offset from a position of the reference value by an amount indicated by the reference vector; and

generating a filtered array representing an improved version of the first frame, wherein the filtered array contains pixel values that are combinations of pixel values from the first array and the reference values, and wherein each pixel value in the filtered array is equal to a corresponding pixel value in the first array if a difference between the corresponding value and a corresponding reference value in the reference array is greater than a threshold value and is equal to a linear combination of the corresponding pixel value and the corresponding reference value if the difference is not greater than the threshold value.

17. The method of claim 16, wherein determining reference vectors comprises combining of the motion vector for the first area containing the pixel and motion vectors for first areas that are nearest to the first area containing the pixel.

18. A method for improving appearance of an image, comprising:

representing the image using a first array of pixel values;

determining a first range for pixel values in a first block that is in the first array and includes a target pixel value;

determining a second range for pixel values in a second block that is in the first array and includes the target pixel value, wherein the second block is smaller than the first block;

selecting a spatial filter from a plurality of spatial filters, wherein the spatial filter is selected according to the first and second ranges; and

applying the selected spatial filter to the target pixel value, wherein applying the selected spatial filter combines the target pixel value with surrounding pixel values in the first array to generate a corresponding pixel value in a second array representing the image.

19. The method of claim **18**, wherein the second block is a 3x3 block of pixel values centered on the target pixel value.

20. The method of claim **19**, further comprising performing an inverse frequency transformation on a block of transform coefficients to determine the pixel values in the first block.

21. The method of claim **18**, wherein selecting the spatial filter comprises:

selecting a first spatial filter in response to the second range being greater than a first threshold value; and

selecting a second spatial filter in response to the first range being greater than a second threshold and the second range being less than a third threshold, wherein the second spatial filter is stronger than first spatial filter.

22. The method of claim **18**, for each pixel value in the first array, using that pixel value as the target pixel in a repetition of the steps of determining the first range, determining the second range, selecting a spatial filter, and applying the selected spatial filter.

23. The method of claim **18**, wherein applying the selected spatial filter comprises:

identifying a likeness threshold that corresponds to the second range; and

excluding from the combination that generates the corresponding pixel value any pixel values that differ from the target pixel value by more than the likeness threshold.

24. The method of claim **18**, wherein selecting a spatial filter comprises selecting a filter strength parameter β corresponding to the first and second ranges.

25. The method of claim **24**, wherein:

the target pixel value is Pij;

the corresponding value is Oij and is determined from pixel values of the first array according to an equation Oij=(1=β)*Pij+β*F(Pij), where F(Pij) is a linear combination of one or more pixel values near the target pixel value in the first array.

26. The method of claim **25**, wherein applying the selected spatial filter comprises identifying a likeness threshold that corresponds to the second range, and linear combination F(Pij) excludes pixel values that differ from the target pixel value by more than the likeness threshold.

27. A method for improving appearance of an image, comprising:

representing the image using a first array of pixel values;

determining a range of pixel values in a block that is in the first array and includes a target pixel value;

identifying a likeness threshold that corresponds to the range determined; and

generating an output pixel value for a second array representing an improved-appearance version of the image, the output pixel value being a linear combination of the target pixel value and one or more pixel values of the first array, the linear combination excluding pixel values that differ from the target pixel by more than the likeness threshold.

28. The method of claim **27**, wherein the likeness threshold is linearly related to the range.

29. A method for improving appearance of a video image, comprising:

decoding a signal to generate a first series of arrays of pixel values, wherein each array of pixel values represents a frame in the video image and comprises a set of blocks;

applying a block boundary filter to pixel values at boundaries of the blocks in the frames to generate a second series of arrays of pixel values, wherein applying the block boundary filter leaves unchanged pixel values that are not at a boundary of any of the blocks;

performing a temporal filtering operation that combines pixel values from different arrays in the second series to generate a third series of arrays of pixel values; and

applying a spatial filter to the arrays in the third series to generate a fourth series of arrays representing the video image with improved appearance.

30. The method of claim **29**, wherein the signal comprises a plurality of sets of transformation coefficients with each set corresponding to a different one of the blocks in the arrays of the first series, and decoding comprises for each set of transformation coefficients, performing an inverse transformation on the set of transformation coefficients to generate pixel values in the block corresponding to the set of transformation coefficients.

31. The method of claim **29**, wherein applying the spatial filter comprises:

filtering each pixel value in an array using a filter that has an adjustable parameter; and

altering the parameter according to content of an area in a frame that includes a pixel represented by a pixel value being filtered.

* * * * *

US005621468A

United States Patent [19]

Kim

[11] Patent Number: 5,621,468

[45] Date of Patent: Apr. 15, 1997

[54] **MOTION ADAPTIVE SPATIO-TEMPORAL FILTERING OF VIDEO SIGNALS**

[75] Inventor: **Jong-Hoon Kim**, Seoul, Rep. of Korea

[73] Assignee: **Daewoo Electronics Co., Ltd.**, Seoul, Rep. of Korea

[21] Appl. No.: **320,702**

[22] Filed: **Oct. 7, 1994**

[51] **Int. Cl.**[6] **H04N 7/32**

[52] **U.S. Cl.** **348/416**; 348/699

[58] **Field of Search** 348/384, 390, 348/396, 400–402, 407, 409–413, 415, 416, 420, 429, 607, 699; 382/232, 236, 238, 244; H04N 7/130, 7/137

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,694,342	9/1987	Klees	348/607
4,745,458	5/1988	Hirano et al.	348/429
4,771,331	9/1988	Bierling et al.	348/396
4,873,573	10/1989	Thomas et al.	348/416
5,260,782	11/1993	Hui	348/416
5,280,350	1/1994	DeHaan et al.	348/699
5,311,310	5/1994	Jozawa et al.	348/699

Primary Examiner—Tommy P. Chin
Assistant Examiner—Richard Lee
Attorney, Agent, or Firm—Anderson Kill & Olick P.C.

[57] **ABSTRACT**

A motion adaptive spatio-temporal filtering method is employed as a prefilter in an image coding apparatus, which processes the temporal band-limitation of the video frame signals on the spatio-temporal domain along the trajectories of a moving component without temporal aliasing by using a filter having a band-limitation characteristic according to a desired temporal cutoff frequency and the velocity of moving components.

4 Claims, 6 Drawing Sheets

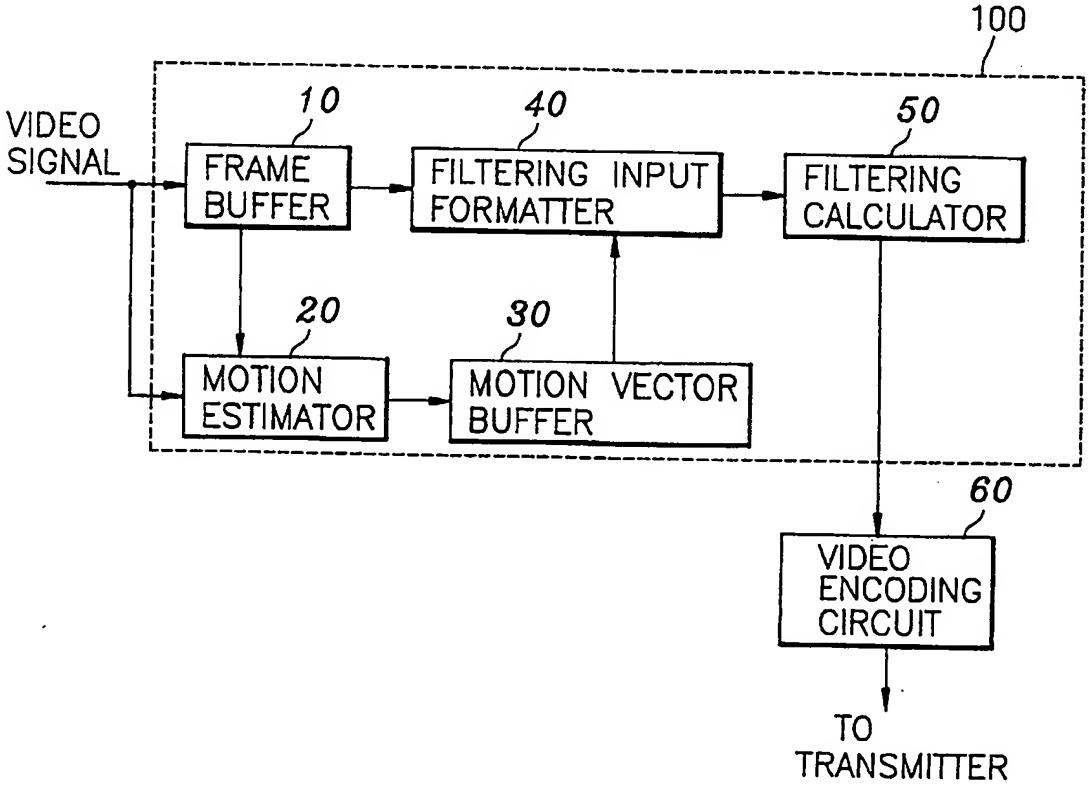
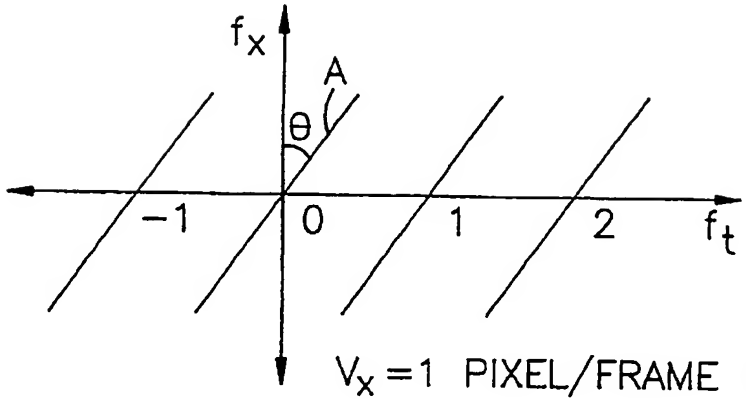
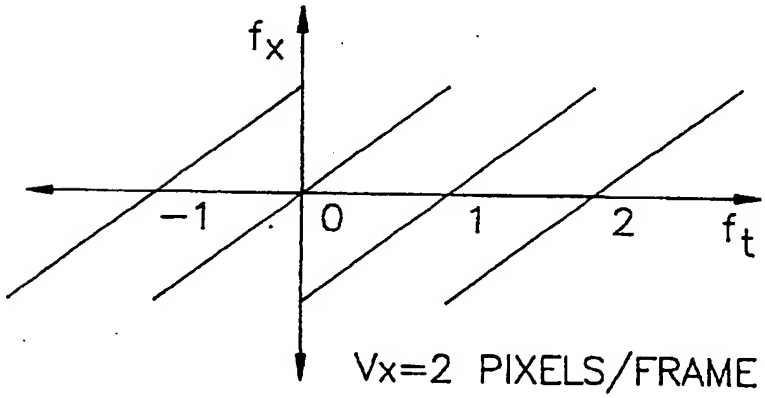


FIG. 1A



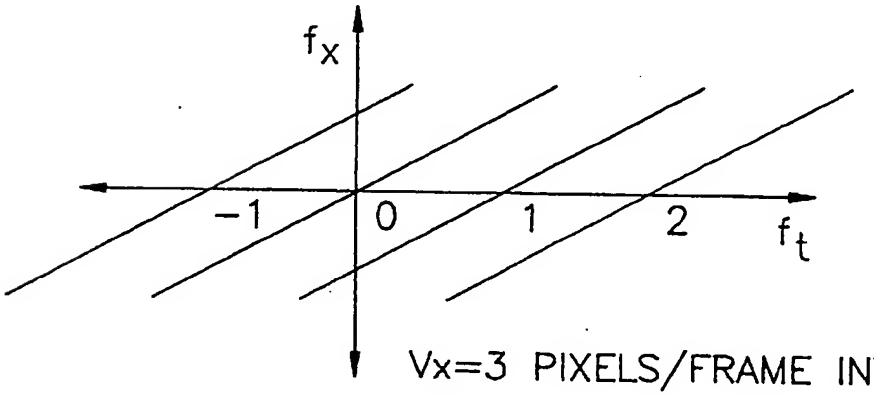
$V_x = 1$ PIXEL/FRAME INTERVAL

FIG. 1B



Vx=2 PIXELS/FRAME INTERVAL

FIG. 1C



Vx=3 PIXELS/FRAME INTERVAL

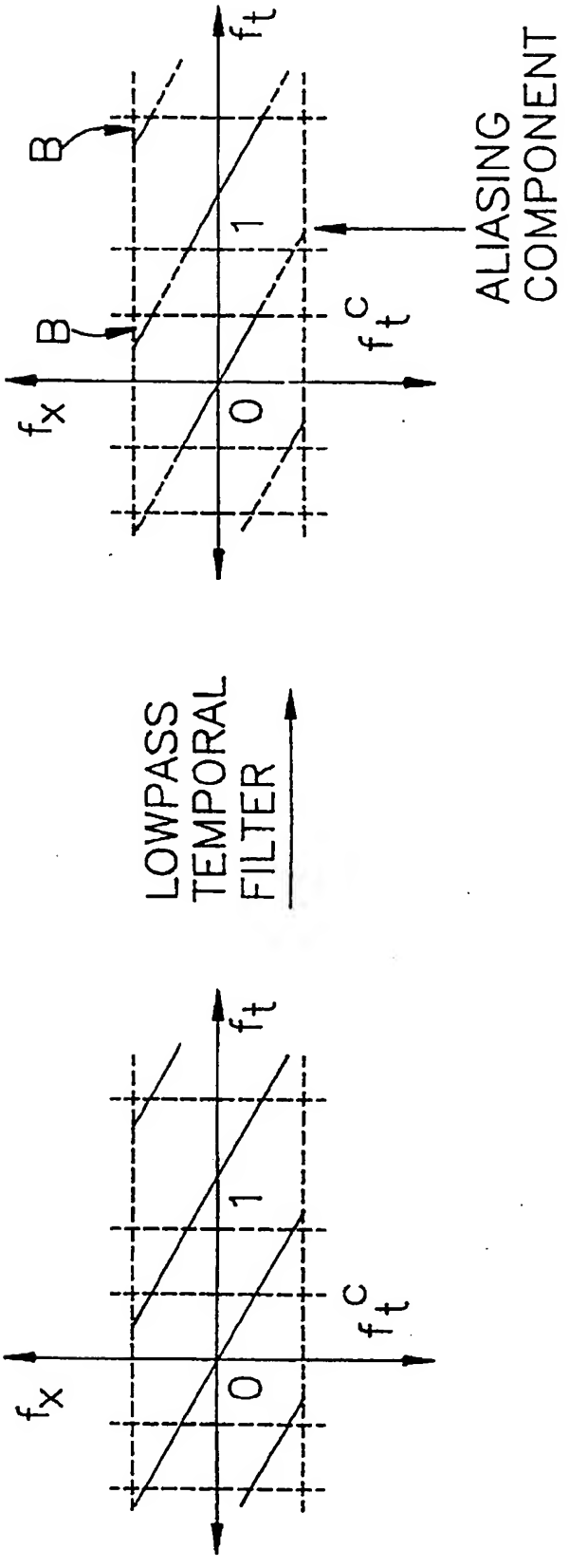

f_t
B
B
1
0
f_x
f_t^c
ALIASING COMPONENT
LOWPASS TEMPORAL FILTER
f_t
1
0
f_x
f_t^c

FIG. 2

FIG. 3

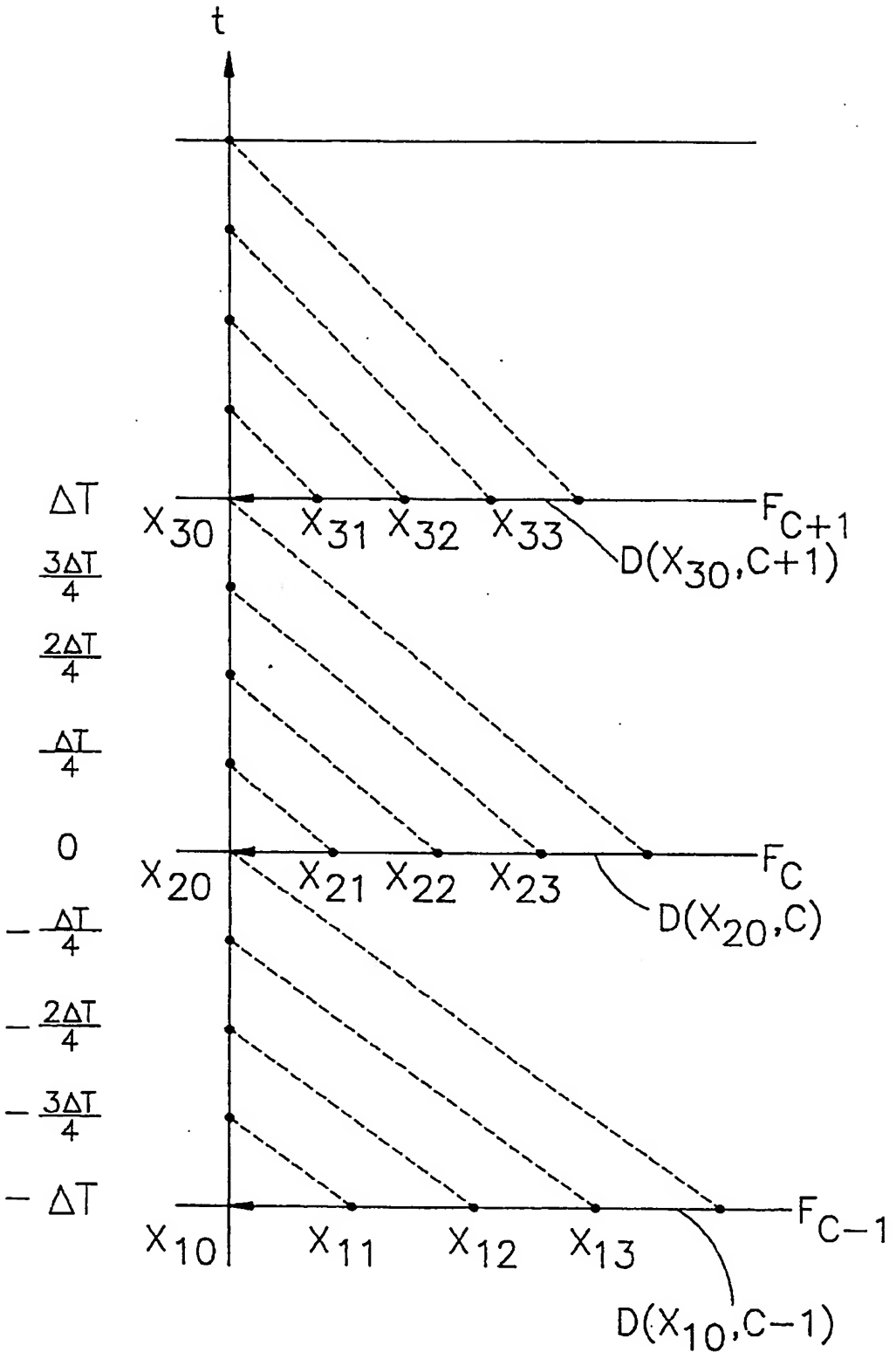

t
ΔT
3ΔT/4
2ΔT/4
ΔT/4
0
−ΔT/4
−2ΔT/4
−3ΔT/4
−ΔT
X30
X31
X32
X33
FC+1
D(X30,C+1)
X20
X21
X22
X23
FC
D(X20,C)
X10
X11
X12
X13
FC−1
D(X10,C−1)

FIG. 4A

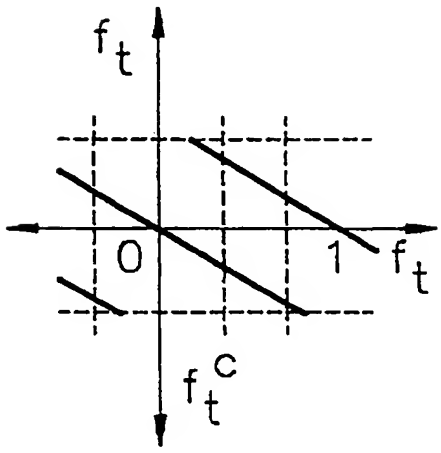


FIG. 4B

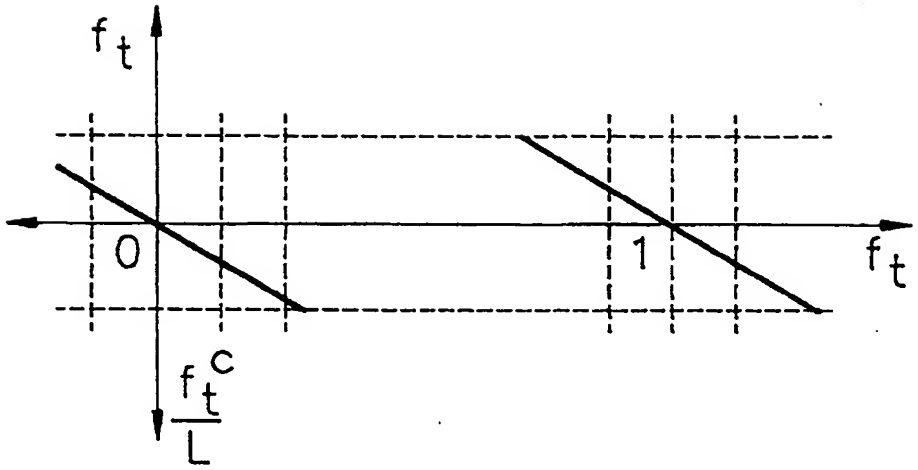


FIG. 4C

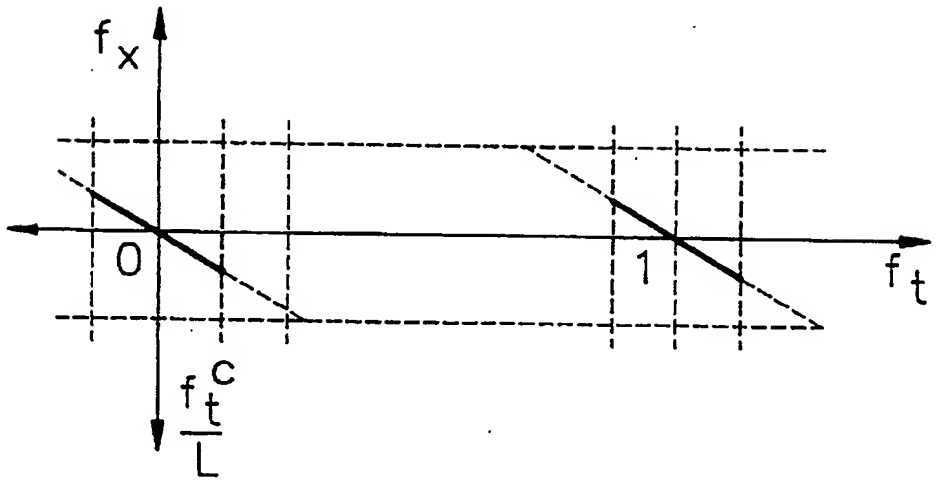


FIG. 4D

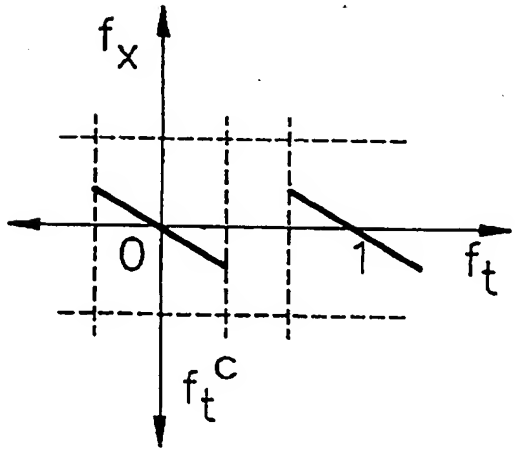
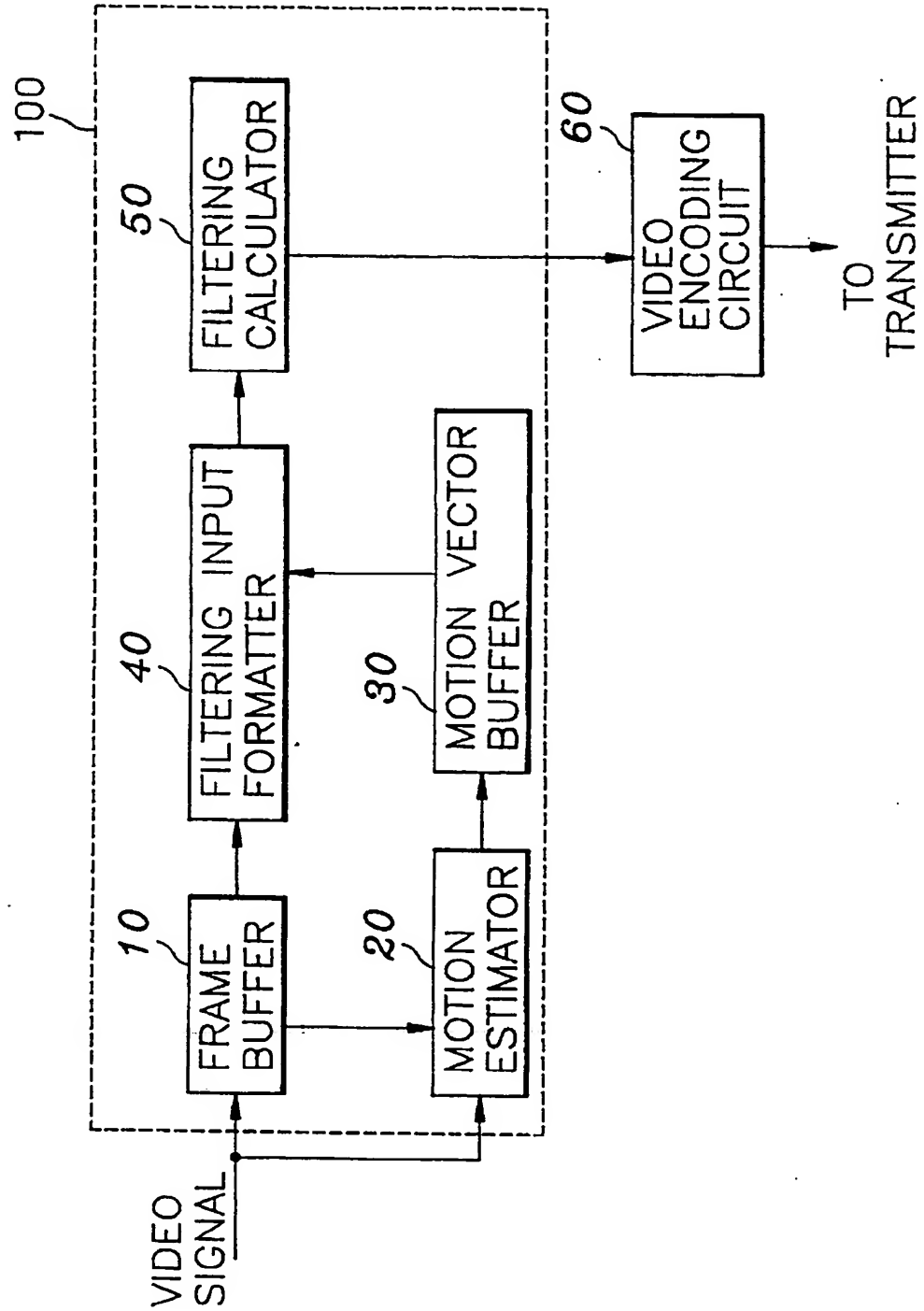


FIG. 5


100
FILTERING CALCULATOR
50
FILTERING INPUT FORMATTER
40
MOTION VECTOR BUFFER
30
FRAME BUFFER
10
MOTION ESTIMATOR
20
VIDEO SIGNAL
60
VIDEO ENCODING CIRCUIT
TO TRANSMITTER

MOTION ADAPTIVE SPATIO-TEMPORAL FILTERING OF VIDEO SIGNALS

FIELD OF THE INVENTION

The present invention is directed to a method and an apparatus for the temporal filtering of video signals; and, in particular, to a motion adaptive spatio-temporal filter-(MASTF) for use in an image encoding apparatus, capable of achieving a temporal band limitation without incurring temporal aliasing effects and thereby obtaining an improved picture quality.

DESCRIPTION OF THE PRIOR ART

In digital television systems such as video-telephone, teleconference and high definition television systems, an image coding apparatus has been used to reduce a large volume of data defining each frame of video signals by way of employing various data compression techniques, for example, a transform coding using a Discrete Cosine Transform, and a motion compensation coding for reducing the temporal relationship between two successive frames.

In order to effectively carry out the data compression process, most real-time image coding apparatus available in the art employ various filters as a part of a front-end processing for the filtering and frame rate reduction. These filters serve to eliminate or alleviate temporal noises and perform band limitation to thereby improve the picture quality and coding efficiency.

One of such prior art apparatus is disclosed in an article by Eric Dubois et al.,"Noise Reduction in Image Sequences Using Motion-Compensated Temporal Filtering ", *IEEE Transactions on Communications, COM*-32, No. 7 (July, 1984), which utilizes a nonlinear recursive temporal filter to reduce noise components which may arise in an initial signal generation and handling operation. This temporal filter employs a motion compensation technique to perform the filtering in the temporal domain along the trajectory of a motion to thereby reduce noise components in moving areas without modifying the details of an image.

Another prior art apparatus is described in an article by Wen-Hsiung Chen et al., "Recursive Temporal Filtering and Frame Rate Reduction for Image Coding", *IEEE Journal on Selected Areas in Communications SAC*-5 (August, 1987), which also employs a recursive temporal filter to perform a recursive filtering and frame rate reduction. This filter when applied in the temporal domain can smooth out frame-to-frame input noises and improve the picture quality.

U.S. Pat. No. 4,694,342 issued to K. J. Klees provides an apparatus which utilizes a spatial filter that can function both recursively and non-recursively for removing noises from a video image while substantially preserving the details thereof. This filter includes a lookup table for storing predefined and filtered output pixel values and predefined feedback pixel values wherein certain portions of an incoming image are filtered non-recursively to substantially preserve the image details while certain other portions of the same image are filtered recursively to remove noises therefrom.

While the above and other prior art apparatus may be capable of reducing the noises in moving areas without altering the image details through the use of a lowpass filtering technique performed along the trajectory of a motion, such approaches tend to introduce artifacts in those areas where the motion occurs in a relatively high speed. As a result, such apparatus are not equipped to adequately deal with the temporal band limitation or the visual artifacts resulting from temporal aliasing.

If the repeated spectra include the aliasing components, visual artifacts appear in the image. Especially, those moving areas comprised of spatial high frequency components may distort psychovisual effects: this is, the perceived velocity on moving areas may differ from the actual velocity. To achieve an efficient temporal band-limitation, therefore, it is desirable to have a temporal filter which is not affected by the aliasing effect.

SUMMARY OF THE INVENTION

It is, therefore, a primary object of the present invention to provide a motion adaptive spatio-temporal filtering method capable of effectively performing temporal band-limitation of a video signal without incurring temporal aliasing and thereby improving the picture quality.

In accordance with the present invention, there is provided a method for filtering a video signal with a predetermined temporal cutoff frequency to achieve a temporal band-limitation thereof, wherein said video signal includes a multiplicity of frames each of which having a multiple number of pixels, the method for obtaining filtered result for a target pixel in a target frame in the video signal which comprises the steps of:

estimating a multiplicity of motion vectors each of > which represents the movement at the target pixel position in each frame of the video signal;

determining, as a filtering input function, a multiplicity of groups of pixel values on trajectories of the target pixel wherein each of the groups is determined on the trajectory of the target pixel in a corresponding frame through the use of the motion vector for the frame; and

performing a convolution of the filtering input function with a predetermined filter impulse response, thereby obtaining a filtered video signal which has the predetermined temporal bandwidth without temporal aliasing.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects and features of the instant invention will become apparent from the following description of preferred embodiments taken in conjunction with the accompanying drawings, in which:

FIGS. 1A, 1B and 1C are diagrams illustrating base-band spectrum distributions as a function of the velocity of a moving object;

FIG. 2 is a diagram depicting a result of the conventional lowpass filtering in the temporal domain with a fixed temporal cutoff frequency;

FIG. 3 is a diagram for illustrating a filtering input function in the spatio-temporal domain;

FIGS. 4A to 4D illustrate the result of the motion adaptive spatio-temporal filtering in accordance with the present invention; and

FIG. 5 is a schematic block diagram representing an image coding apparatus employing the motion adaptive spatio-temporal filtering method in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A video signal may be treated in terms of its 3-dimensional, i.e., horizontal, vertical and temporal components;

and described as a continuous function $f_3(x,y,t)$. Assuming that its moving objects have only a constant-velocity rigid translational motion $v=(v_x,v_y)$, the Fourier transform of the continuous video signal, $F_3(\cdot)$, may be represented as follows:

$$F_3(f_x, f_y, f_t)=F_2(f_x, f_y)\cdot\delta(f_xv_x+f_yv_y+f_t) \quad \text{Eq. (1)}$$

wherein $F_2(f_x, f_y)$ is the Fourier transform of a 2-dimensional video signal $f_2(x,y)$, and $\delta(f_xv_x+f_yv_y+f_t)$ represents a tilted plane in a 3-Dimensional frequency space described by the equation $f_xv_x+f_yv_y+f_t=0$ so that the baseband exists only on a 2-Dimensional frequency plane. Eq. (1) is disclosed in, e.g., an article by R. A. F. Belfor, et al., "Motion Compensated Subsampling of HDTV", *SPIE*, 1605, *Visual Communications and Image Processing* '91, pp 274–284 (1991). From the location of a baseband spectrum, a spatio-temporal bandwidth can be anticipated. That is, if a temporal bandwidth f_t^w is given, the relationship among the temporal bandwidth f_t^w, the spatial bandwidth f_x^w and f_y^w, and the velocity components v_x and v_y is obtained from Eq. (1) as follows:

$$f_t^w=f_x^w\cdot v_x+f_y^w\cdot v_y \quad \text{Eq. (2)}$$

wherein f_x^w and f_y^w are the respective spatial bandwidth components in x and y directions. From Eq. (2), it can be seen that the temporal bandwidth is proportional to the velocity of the moving objects; and when the temporal bandwidth is fixed, the spatial bandwidth becomes inversely proportional to the velocity of the moving object.

Since the video signal for the filtering is sampled with a spatial and temporal sampling frequencies, the sampled video signal is represented as 3-Dimensional sampled data, i.e., pixels. Therefore, sampling of the continuous function $f_3(\cdot)$ may be expressed by multiplying the continuous function $f_3(x,y,t)$ with a 3-Dimensional array of delta functions. A spectrum distribution of the pixels may be then given by the convolution of Fourier transform of $f_3(\cdot)$ and a delta function. As a result, the spectrum of the pixels is replicated at intervals of the sampling frequencies by the characteristics of the delta function.

Referring first to FIGS. 1A, 1B, and 1C, there are shown baseband spectrum distributions as a function of the velocity of a moving object v_x=1 pixel/frame interval, v_x=2 pixels/frame interval and v_x=3 pixels/frame interval, wherein solid lines indicate the replicas of a baseband; and the temporal sampling frequency is normalized to 1; and the spatial (x axis direction) and temporal frequencies are designated as f_x and f_t, respectively.

The motion of a pixel A in the moving object causes the spectrum to become skewed from the spatial frequency axis as shown in FIG. 1A. As shown in FIGS. 1A, 1B and 1C, the angle θ of said skewing increases as does the velocity. From Eq. (2), the reason for the skewing can be readily understood by considering the temporal frequency at a pixel in the video signal: since the spectrum distribution on the spatio-temporal frequency domain is related to the product of the spatial frequency and the speed of the moving object, a higher velocity of the moving object gives rise to a higher temporal frequency. It should be stressed that the spectrum is skewed and not rotated.

Referring to FIG. 2. results of lowpass filtering in the temporal domain with a fixed temporal cutoff frequency f_t^c are illustrated. In order to perform the temporal filtering, two assumptions may be made as follows: first, baseband spectrum has no spatial aliasing components, and secondly, for the sake of simplicity, therm exists only purely horizontal motion (represented in terms of f_x) with a constant velocity. In FIG. 2, the filtered result contains, e.g., spatial high frequency components B of adjacent spectra which represent temporal aliasing. That is, the spatial high frequency components affect the temporal low frequency components of the adjacent replicas. In other words, a disturbance between the spatial high frequency components and the low frequency ones of the adjacent replicas appears in the displayed image.

As may be seen from Eqs. (1) and (2), the relation between the spatial (including the vertical and the horizontal components) and temporal frequencies f_s and f_t are represented as follows:

$$f_s=\frac{1}{|v|}\cdot f_t \quad \text{Eq. (3)}$$

wherein the spatial frequency f_s is defined on f_x-f_y plane. As is seen from Eq. (3), it should be appreciated that, when the temporal cutoff frequency is fixed in order to limit the temporal bandwidth, the spatial cutoff frequency becomes inversely proportional to the absolute value of the velocity of the moving object.

Assuming that $h(\cdot)$ is an impulse response of a lowpass temporal filter and, for simplicity, there exists only a purely horizontal motion (x axis direction), then the temporal band-limited video signal $g(x,t)$ may be represented as follows:

$$g(x,t)=\int_{-\infty}^{\infty} h(\tau)\cdot f(x,t-\tau)d\tau \quad \text{Eq. (4)}$$

wherein a linear phase filter is used to reduce the effect of a group-delay of a filter response. From the assumption of constant-velocity rigid translational motion $v=(v_x, v_y)$ and purely horizontal motion, a filtering input function may be represented as follows.

$$f(x,t-\tau)=f(x+v_x\tau,t) \quad \text{Eq. (5)}$$

From Eq. (5), the displacement of the moving pixel along the temporal frequency axis can be represented by its trajectory in the spatial domain at a point on the temporal axis. Thus, Eq. (4) may be rewritten as:

$$g(x,t)=\int_{-\infty}^{\infty} h(\tau)\cdot f(x+v_x\tau,t)d\tau \quad \text{Eq. (6)}$$

On the other hand, in case of a real video signal the assumption of constant-velocity ligid translational motion is not always valid. Furthermore, in the case that there is no moving object, each pixel value of the video data signal vary with the time due to, e.g., changes in lighting source and characteristics of video signal generating device such as a video camera. In such cases, Eq. (5) holds true only for a short period of time and can be rewritten as:

$$f(x,t-(k+1)\Delta t)=f(x+v_x)(t-k\Delta t)\cdot\Delta t,\ t-k\Delta t) \quad \text{Eq. (7)}$$

wherein Δt denotes a short period of time, e.g., a frame interval and k is an integer. In accordance with Eq. (7), the equation (6) can be rewritten as:

$$g(x,t) = \sum_{k=-\infty}^{\infty} \int_{k\Delta t}^{(k+1)\Delta t} h(\tau) f(x + v_x(t-k\Delta t)\cdot(\tau - k\Delta t), t - k\Delta t) d\tau \quad \text{Eq. (8)}$$

From Eq. (8), it can be appreciated that the temporal filtering of Eq. (4) can be achieved by spatio-temporal filtering with its filtering input function f(·).

Eq. (8) is a continuous description of the motion adaptive spatio-temporal filtering. Similar results hold in the discrete case: the integral is replaced by summation and dτ is represented by Δτ and j. Eq. (8) is then given by

$$g(x,n) = \sum_{j=-N}^{N} \left\{ \sum_{l=0}^{L-1} h(Lj+l) \cdot f(x + v(x,n-j)\cdot\Delta\tau \cdot l, n-j) \right\} \quad \text{Eq. (9)}$$

wherein n is a frame index; the velocity and the filtering positions are replaced by vectors v and x; filter impulse response h(·) comprising (2N++1)×L filter coefficients is predetermined in conjunction with the temporal cutoff frequency and the predetermined numbers N, L(N,L are positive integers); and if we denote a pixel-to-pixel interval as Δx, Δτ is selected to satisfy $|v(\cdot)\cdot\Delta\tau| \leqq |\Delta x|$ (If Δτ fails to satisfy the condition, it may cause spatial aliasing).

Therefore, as may be seen from Eq. (9), the temporal band-limitation can be achieved by sparío-temporal filtering, i.e., lowpass filtering of the filtering input function taken from both spatial and temporal domains.

On the other hand, if ΔT is a frame to frame interval, then LΔr is equal to ΔT and v(·)·ΔT is equal to D(·) which is a motion vector representing a displacement of a pixel between two neighboring frames. Then, Eq.(9) can be modified as follows:

$$g(x,n) = \sum_{j=-N}^{N} \left\{ \sum_{l=0}^{L-1} h(Lj+l) \cdot f\left(x + D(x,n-j)\cdot\frac{l}{L}, n-j \right) \right\} \quad \text{Eq. (10)}$$

wherein L is selected to satisfy $|D(\cdot)| \leqq |\Delta x|\cdot L$ (This condition is equivalent to the condition $|v(\cdot)\cdot\Delta\tau| \leqq |\Delta x|$ described earlier, therefore if L fails to satisfy this condition, it may cause spatial aliasing). Eq. (10) is an implementation of Eq. (9). The temporal band-limitation is achieved by spatio-temporal filtering, i.e., lowpass filtering on the filtering input function f(·) which comprises a multiplicity of, e.g., (2N+1), groups of filtering input data wherein each group includes a predetermined number of, e.g, L filtering input data which are obtained from pixel values of corresponding frame in the video signal. In Eq. (10), (x+D(x, n−j)·l/L) which denotes a position of filtering input data in (n−j)th frame of the video signal, may not coincide with exact pixel positions. In that case, the filtering input data can be determined from adjacent pixels located around the position by using, e.g., bilinear interpolation method which determines a weighted sum of the adjacent pixel values as the filtering input data. That is, the filtering input function is obtained on the sparío-temporal domain along the trajectories of moving object. Specifically, a group of input data included in the filtering input function f(·) may be determined from the pixel values of a corresponding frame using the motion vector which represents the displacement of the moving object between the frame and its previous frame in the video signal as will be described in conjunction with FIG. 3.

On the other hand, the filter impulse response comprising a plurality, i.e., (2N+1)×L, of filter coefficients serves to limit the bandwidth of the video signal to a predetermined bandwidth. These filter coefficients may be predetermined based on a desired temporal cutoff frequency and a predetermined numbers N and L. For example, when the temporal cutoff frequency is f_t^c, the filter impulse response is designed with a spatio-temporal cutoff frequency of f_t^c/L.

Actually, as may be seen from Eq. (10), the filtered data g(·), i.e., band-limited data, is obtained by convolving each group of filtering input data with corresponding filter coefficients and by summing each group of filtered input data.

Referring to FIG. 3, there is shown an explanatory diagram illustrating the filtering input function for the motion adaptive spatio-temporal filtering method of the present invention. For the sake of simplicity, each frame is denoted as a line, e.g., F_{c-1}, F_c and F_{c+1}, and N and L of Eq. (10) are assumed to be 1 and 4, respectively. In other words, to obtain the filtered data for a target pixel in a target frame F_c, three filtering input frames, i.e., the target frame F_c containing the target pixel to perform filtering operation thereon and its two neighboring frames F_{c-1}, F_{c+1}, are used for the filtering process wherein c−1, c, and c+1 denote frame indices; and four filtering input data are determined on each filtering input frame based on the motion vector for the pixel at the target pixel position in its subsequent frame. The position of the target pixel is denoted as x_{10}, x_{20} and x_{30} in the frames F_{c-1}, F_c and F_{c+1}, respectively, and the vertical axis is a time axis.

In order to obtain the filtered data for the target pixel at x_{20} in the target frame F_c, a multiplicity of, i.e., three, groups of filtering input data are decided, each group including a predetermined number, e.g., 4, of filtering input data located on the corresponding motion trajectory for the target pixel in the corresponding filtering input frame. Specifically, three groups of filtering input data positioned at (x_{10}, X_{11}, X_{12}, X_{13}), (x_{20}, X_{21}, X_{22}, x_{23}) and (x_{30}, x_{31}, x_{32}, x_{33}) are determined on the trajectories of the pixels at the target pixel position based on the motion vectors D(x_{10}, c−1), D(x_{20}, c) and D(x_{30}, c+1) in the frames F_{c-1}, F_c and F_{c+1}, respectively.

As shown in FIG. 3, it is readily appreciated that the filtering input data are equivalent to the target pixel values in temporally interpolated or upsampled frames of the video signal. For instance, the filtering input data at x_{11} in the frame F_{c-1} is equivalent to the pixel value at x_{10} at time t=−3ΔT/4. That can be denoted as:

$$f\left(x_{10} + D(x_{10}, -\Delta T)\cdot\frac{1}{4}, -\Delta T \right) = f\left(x_{10}, -\frac{3}{4}\Delta T \right) \quad \text{Eq. (11)}$$

The equivalence between the spatial domain and the time domain is denoted as dotted line in FIG. 3.

Referring now to FIGS. 4A to 4D, there is shown the result of the lowpass temporal filtering of the video signal on a spatio-temporal domain through the use of the motion adaptive sparío-temporal filtering method. In FIG. 4A, there is shown a baseband spectrum of the original video signal. As described above, the process of obtaining each group of filtering input data is equivalent to temporal upsampling or interpolating as illustrated in FIG. 4B. If the desired cutoff frequency of the temporal lowpass filtering is f_t^c, the spatio-temporal cutoff frequency f_c of the filter of the present invention is f_t^c/L as shown in FIG. 4C. The final spectra for the filtered results are shown in FIG. 4D which are the subsampled versions of the spectra in FIG. 4C(note that the filtered results are not provided for the interpolated frames). Comparing with the temporal band-limitation depicted in FIG. 2, it should be readily appreciated that the spatio-temporal band-limitation of the present invention is not affected by temporal aliasing components.

As may be seen from the Eq. (10) and FIGS. 3, 4A, 4B, 4C, and 4D, it should be appreciated that the filtering

operation is performed on a spario-temporal domain along the trajectory of moving objects to thereby achieve a temporal band limitation. Therefore, the temporal aliasing, which may occur in the repeated Spectra when the velocity of the moving objects is increased, can be effectively eliminated by the inventive filter to thereby greatly reduce the visual artifacts appearing in the moving areas in the image.

Referring now to, FIG. **5**, there is shown an image coding apparatus employing the motion adaptive spatio-temporal filter in accordance with a preferred embodiment of the present invention. The image coding apparatus comprises a filtering circuit **100** for performing the motion adaptive spatio-temporal filtering in accordance with the present invention and a video encoding circuit **60** for eliminating redundancies in the filtered video signal in order to compress the video signal to a more manageable size for the transmission thereof. The video signal is generated from a video signal source, e.g., video camera(not shown), and fed to the filtering circuit **100**.

The filtering circuit **100** performs the motion adaptive spatio-temporal filtering operation, as previously described, in accordance with Eq. (10). The filtering circuit **100** includes a frame buffer **10**, a motion estimator **20**, a motion vector buffer **30**, a filtering input formatter **40** and a filtering calculator **50**. The frame buffer **10** stores a current frame which is being inputted to the filtering circuit **100** and a multiplicity of, e.g., (2N+1), previous frames, i.e., filtering input frames to be used in a filtering procedure. Specifically, assuming that N=1, the frame buffer **10** stores the current frame F_{c+2} and three filtering input frames F_{c-1}, F_c and F_{c+1}, wherein F_{c+2}, c+1, c, and c−1 are frame indices. The motion estimator **20** receives two successive frames of the video signal, i.e., the current frame F_{c+2} of the video signal inputted directly from the video signal source and its previous frame F_{c+1} stored in the frame buffer **10**, and extracts motion vectors associated with each of the pixels included in the current frame F_{c+2}. In order to extract motion vectors, various motion estimation method, as well known in the art, may be employed (see, e.g., MPEG Video Simulation Model Three, International Organization for Standardization, Coded Representation of Picture and Audio Information, 1990, ISO-IEC/JTC1/SC2/WG8 MPEG 90/041).

The extracted motion vectors are coupled to the motion vector buffer **30** to be stored therein. In accordance with the present invention, the motion vector buffer **30** stores motion vectors for the frames F_{c+2}, F_{c+1}, F_c and F_{c-1}.

The filtering input frames stored in the frame buffer **10** and the motion vectors associated with the filtering input frames stored in the motion vector buffer **30** are coupled to the filtering input formatter **40**. The filtering input formatter **40** determines a multiplicity, e.g., 3, of groups of filtering input data which constitute the filtering input function f(·) in Eq. (10). As described above, in case filtering input data is determined to be located at a position which does not fall on the exact pixel position, the filtering input formatter **40** provides the filtering input data by calculating a weighted sum of the four neighboring pixels thereof. The filtering input data are coupled to the filtering calculator **50**.

At the filtering calculator **50**, the filtered data g(·) is calculated as represented by Eq. (10) using the filtering input data inputted from the filtering input formatter **40**.

The filter impulse response comprising a plurality of, e.g., (2N+1)×L, filter coefficients is determined according to the desired temporal cutoff frequency f_t^c, N and L which are predetermined so as to satisfy the condition described earlier in conjunction with Eq. (10) by considering the characteristics of the video signal. The filter coefficients may be predetermined prior to the filtering process and stored in the filtering calculator **50**. As described above, the filtering circuit **100** performs the motion adaptive spario-temporal filtering operation to thereby obtain a temporal band-limited video signal.

The filtered video signal outputted from the filtering calculator **50** is coupled to the video encoding circuit **60** wherein the video signal is compressed by various method known in the art (see, e.g., MPEG Video Simulation Model Three, International Organization for Standardization, Coded Representation of Picture and Audio Information, 1990, ISO-IEC/JTC1/SC2/WG8 MPEG 90/041). The encoded video signal is coupled to a transmitter for the transmission thereof.

While the present invention has been shown and describe with reference to the particular embodiments, it will be apparent to those skilled in art that many changes and modifications may be made without departing from the spirit and scope of the invention as defined in the appended claims.

What is claimed is:

1. An apparatus for providing a filtered data for each of pixels of a video signal by filtering the video signal with a predetermined temporal cutoff frequency to achieve a temporal band limitation thereof, wherein said video signal comprises a multiplicity of filtering input frames which include a target frame to perform a filtering operation thereon and a predetermined number of preceding frames and subsequent frames of said target frame, each of the filtering input frames having a multiple number of pixels, comprising:

means for estimating a plurality of motion vectors each of which represents the movement for each of the pixels included in the video signal;

means for determining a filtering input function for a target pixel included in the target frame, wherein the filtering input function includes a multiplicity of groups of filtering input data; each group of the filtering input data is determined on a trajectory of a pixel at the target pixel position in each of the multiplicity of filtering input frames based on a motion vector of the pixel at the target pixel position;

means for performing a convolution of the filtering input function with a filter impulse response determined according to a spatio-temporal cutoff frequency f_c which is represented as:

$$f_c = \frac{f_t^c}{L}$$

wherein f_t^c is the temporal cutoff frequency; and L is a predetermined positive integer related to the velocity of a moving object in the video signal, thereby obtaining filtered data for the target pixel in the target frame.

2. The apparatus of claim **1**, wherein said filtered data is represented as follows:

$$g(x,n) = \sum_{j=-N}^{N} \left(\sum_{l=0}^{l=L-1} h(Lj+l) \cdot f\left(x + D(x,n-j) \cdot \frac{1}{L}, n-j \right) \right)$$

wherein x is the position of the target pixel; n is the index of the target frame in the video signal; the filter impulse response h(·) includes (2N+1)×L filter coefficients; j is a index whose absolute value is not greater than N; N, L are positive integers; and D(·) is a motion vector representing a motion for the target pixel.

3. A method for providing a filtered data for a target pixel in a video signal by filtering the video signal with a predetermined temporal cutoff frequency to achieve a temporal band limitation thereof, wherein said video signal comprises a multiplicity of filtering input frames which include a target frame having the target pixel therein and a predetermined number of preceding frames and subsequent frames of said target frame, each of the filtering input frames having a multiple number of pixels, comprising the steps of:

estimating a multiplicity of motion vectors each of which represents the movement for each of the pixels at the target pixel position in each frame of the video signal;

determining a filtering input function for the target pixel, wherein the filtering input function includes a multiplicity of groups of filtering input data; each group of the filtering input data is determined on a trajectory of a pixel at the target pixel position in each of the multiplicity of filtering input frames based on a motion vector of the pixel at the target pixel position; and

performing a convolution of the filtering input function with a filter impulse response determined according to a spatio-temporal cutoff frequency f_c which is represented as:

$$f_c = \frac{f_t^c}{L}$$

wherein f_t^c is the temporal cutoff frequency; and L is a predetermined positive integer related to the velocity of a moving object in the video signal, thereby obtaining filtered data for the target pixel in the target frame.

4. The method of claim 3, wherein said filtered data is represented as follows:

$$g(x,n) = \sum_{j=-N}^{N}\left(\sum_{l=0}^{l=L-1} h(Lj+l)\cdot f\left(x + D(x,n-j)\cdot\frac{1}{L}, n-j\right)\right)$$

wherein x is the position of the target pixel; n is the index of the target frame in the video signal; the filter impulse response h(·) includes (2N+1)×L filter coefficients; j is a index whose absolute value is not greater than N; N, L are positive integers; and D(·) is a motion vector representing a motion for the target pixel.

* * * * *

US006041068A

United States Patent [19]

Rosengren et al.

[11] Patent Number: 6,041,068

[45] Date of Patent: *Mar. 21, 2000

[54] METHOD AND APPARATUS FOR MULTIPLEXING AND TRANSMITTING AUTONOMOUSLY/INTRA CODED PICTURES ALONG WITH THE MAIN OR I, P, B PICTURES/VIDEO

[75] Inventors: Jürgen F. Rosengren; Ronald W. J. J. Saeijs; Eric H. J. Persoon, all of Eindhoven, Netherlands

[73] Assignee: U.S. Philips Corporation, New York, N.Y.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: 08/909,144

[22] Filed: Aug. 11, 1997

Related U.S. Application Data

[63] Continuation of application No. 08/422,378, Apr. 14, 1995, abandoned.

[30] Foreign Application Priority Data

Apr. 15, 1994 [EP] European Pat. Off. 94201053

[51] Int. Cl.[7] H04J 3/02; H04N 7/12; H04N 9/74; H04N 5/445

[52] U.S. Cl. 370/538; 348/423; 348/584; 348/563; 370/535; 380/10

[58] Field of Search 380/5, 10, 20; 360/60, 33.1; 348/578, 584, 588, 596, 409, 410, 423, 563, 564, 565, 385; 352/70; 386/33, 109, 111–112, 81, 68; 370/537, 538

[56] References Cited

U.S. PATENT DOCUMENTS

3,580,989	5/1971	Banning	178/5.1
5,091,785	2/1992	Canfield et al.	358/183
5,377,266	12/1994	Katta et al.	380/20
5,420,866	5/1995	Wasilewski	370/110.1
5,426,699	6/1995	Wunderlich et al.	380/20
5,450,209	9/1995	Niimura et al.	358/335
5,483,287	1/1996	Siracusa	348/426
5,485,221	1/1996	Banker et al.	348/563
5,504,484	4/1996	Wilson	341/167
5,504,530	4/1996	Obikane et al.	348/413
5,515,107	5/1996	Chiang et al.	348/473
5,515,437	5/1996	Katta et al.	380/20
5,576,902	11/1996	Lane et al.	386/68
5,598,222	1/1997	Lane	348/568
5,600,573	2/1997	Hendricks et al.	364/514
5,600,721	2/1997	Kitazato	380/20
5,631,693	5/1997	Wunderlich et al.	348/7

FOREIGN PATENT DOCUMENTS

0505985	9/1992	European Pat. Off. .
0534139	3/1993	European Pat. Off. .
9417631	8/1994	WIPO .

OTHER PUBLICATIONS

"ISO/IEC CD 13818: Information Technology–Generic Coding of Moving Pictures and Associated Audio Information" 1993 12–01.

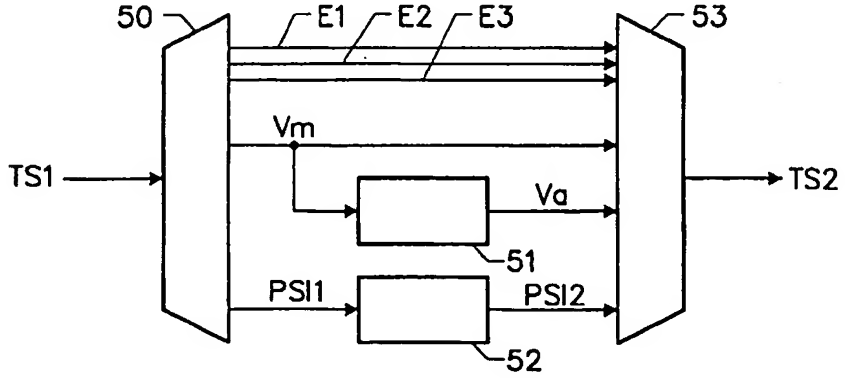
"ISO/IEC CD 13818–2: Information Technology—Generic Coding of Moving Pictures and Associated Audio Information" Part 2: Video 1993 12–01.

Primary Examiner—Douglas W. Olms
Assistant Examiner—David R Vincent
Attorney, Agent, or Firm—Laurie E. Gathman

[57] ABSTRACT

A device for deriving an ancillary signal from a compressed digital video signal (e.g. MPEG), wherein the ancillary signal includes selected parts of the main signal, for example, the DC coefficients of I-pictures, or the unscrambled parts, which parts can be used for display in a (multi-) picture-in-picture television receiver, or as an "appetizer" in order to encourage the user to pay a subscription fee, is described. The ancillary signal can separately be recorded in digital video recorders so as to assist the user in finding the beginning of a scrambled program on tape. The ancillary signal can also be generated at the transmitter end and transmitted at a low bit rate. A decoder for decoding such an ancillary signal is considerably simpler and less expensive than a full-spec MPEG decoder. A decoding method is also described.

22 Claims, 4 Drawing Sheets



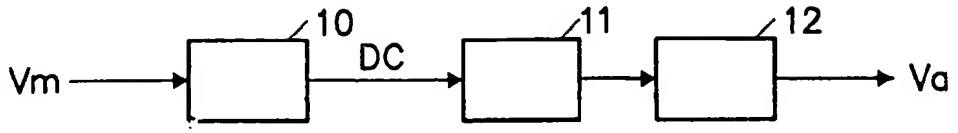


FIG.1

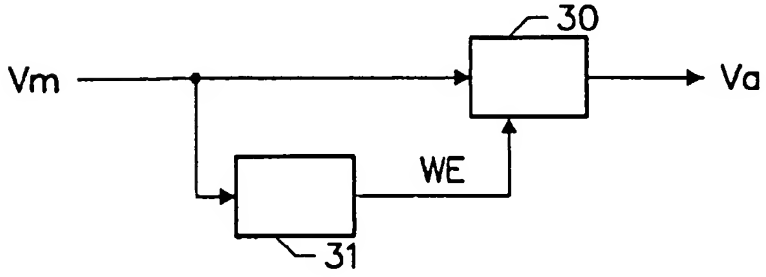


FIG.3

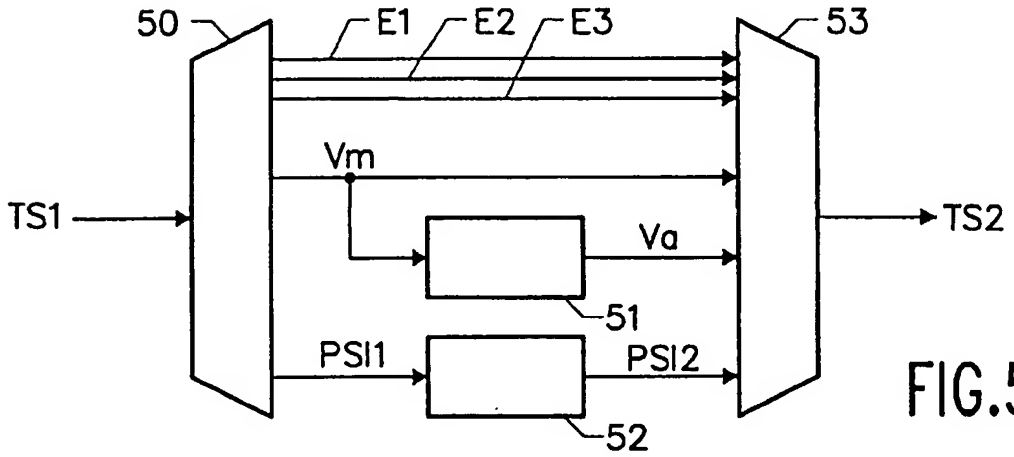


FIG.5

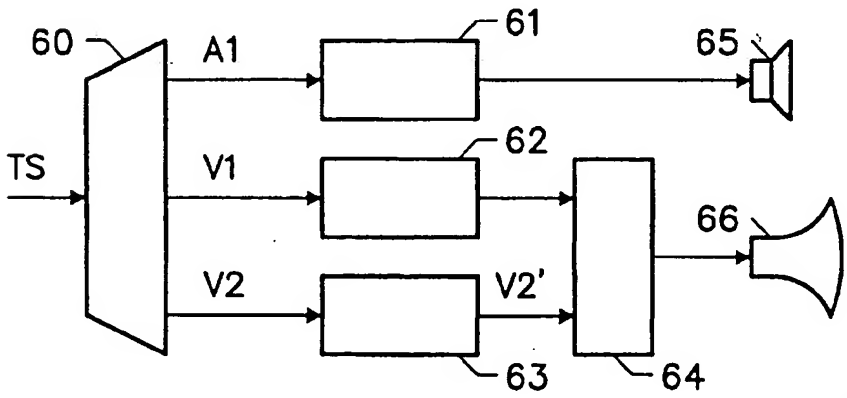


FIG.6

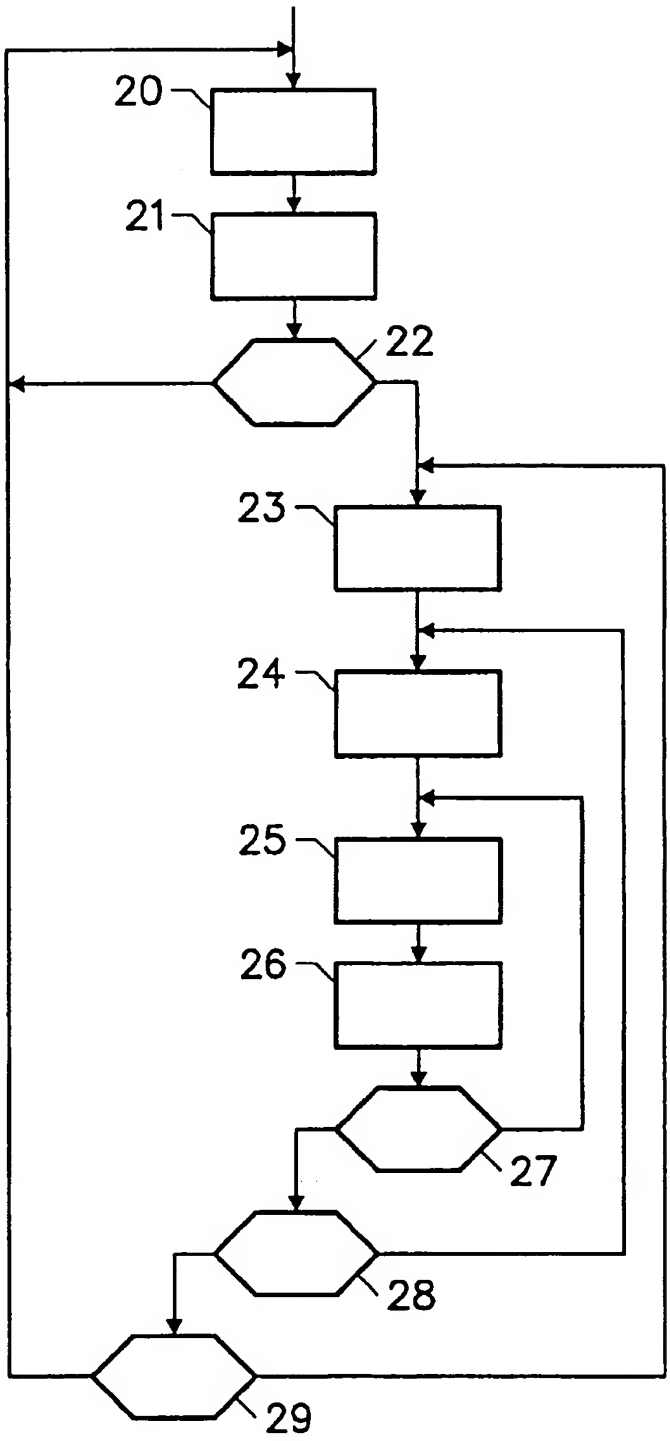


FIG.2

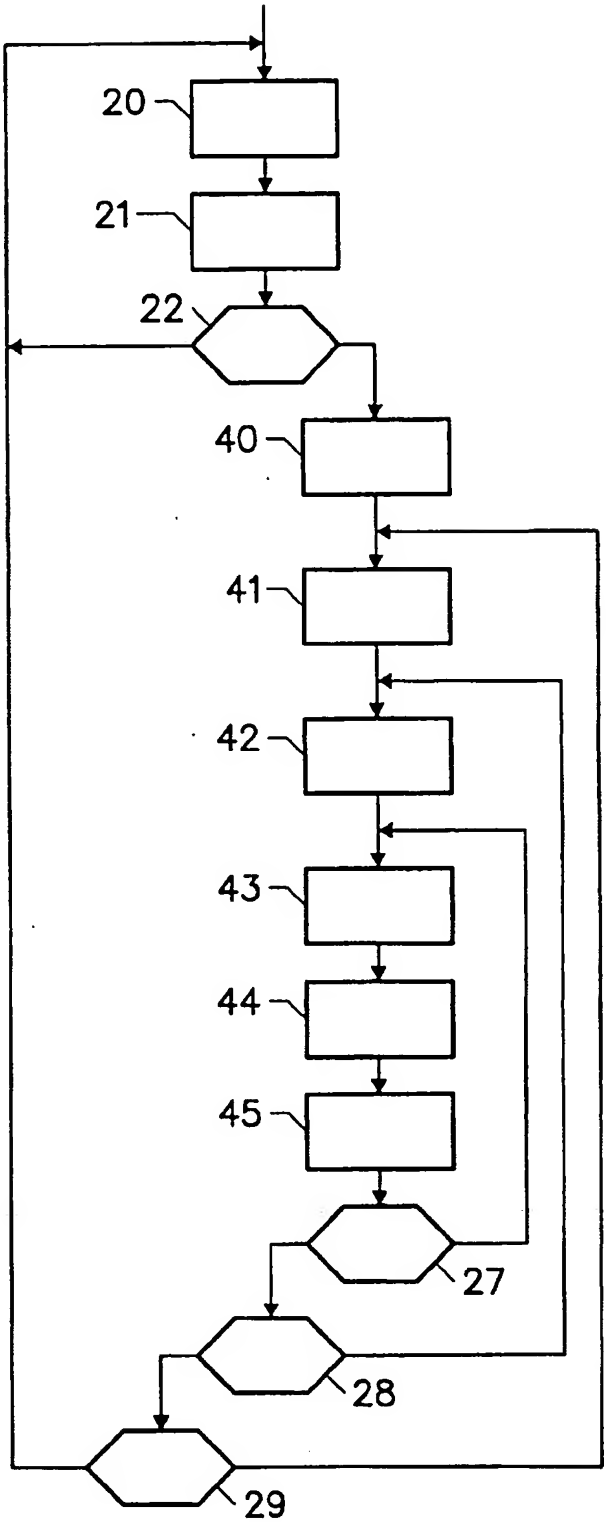


FIG.4

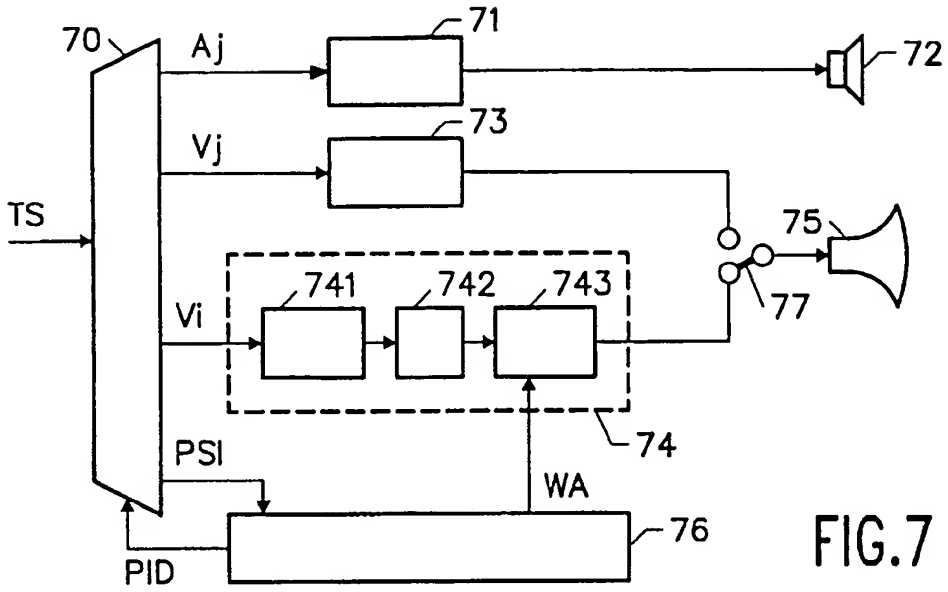


FIG.7

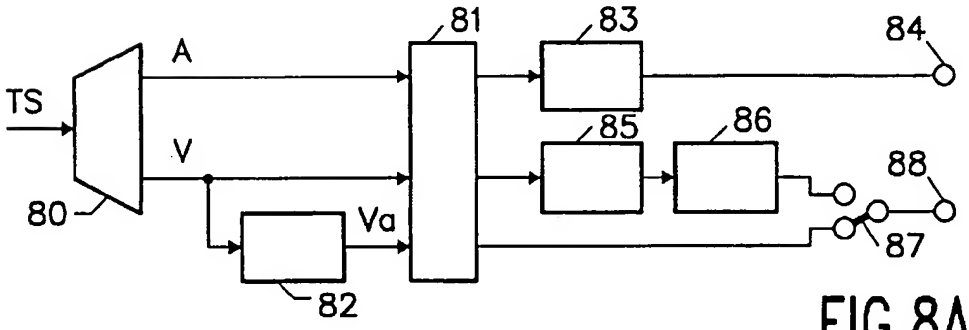


FIG.8A

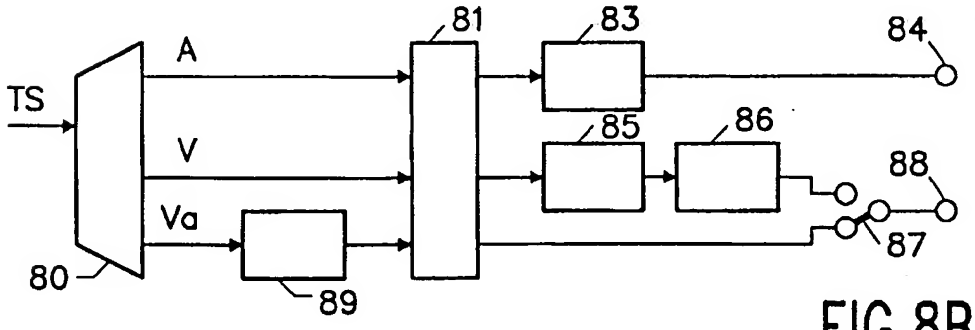


FIG.8B

METHOD AND APPARATUS FOR MULTIPLEXING AND TRANSMITTING AUTONOMOUSLY/INTRA CODED PICTURES ALONG WITH THE MAIN OR I, P, B PICTURES/VIDEO

CROSS REFERENCE TO RELATED APPLICATIONS

This is a continuation of application Ser. No. 08/422,378 filed Apr. 14, 1995 now abandoned.

FIELD OF THE INVENTION

The invention relates to an arrangement for decoding a digital video signal encoded as an MPEG elementary video stream. The invention also relates to television receivers, video recorders, and transmitters comprising such an arrangement.

BACKGROUND OF THE INVENTION

An arrangement for decoding a digital video signal is disclosed in "ISO/IEC CD 13818: Information technology—Generic coding of moving pictures and associated audio information", Dec. 1, 1993, further referred to as the MPEG standard. Part 1 of this standard relates to the system aspects of digital transmission, Part 2 relates more particularly to video encoding.

MPEG2 is a packet-based time multiplex system. Data is transmitted in transport packets. Each transport packet contains data from exactly one elementary stream with which it is associated by means of its packet identifier. Examples of elementary streams are video streams, audio streams, and data streams. One or more elementary streams sharing the same time base make up a program. A typical program might consist of one video stream and one audio stream. One or more programs constitute a transport stream.

OBJECT AND SUMMARY OF THE INVENTION

It is, inter alia, an object of the invention to provide an arrangement which renders it possible to implement new and known features in a more cost-effective manner.

According to the invention, the arrangement for decoding an MPEG elementary video stream is characterized in that the arrangement comprises means for decoding selected parts of said elementary stream, and means for rearranging said selected parts so as to constitute an ancillary video signal. As only selected parts of the elementary signal are decoded, the arrangement is considerably simpler and less expensive than a full-spec MPEG decoder.

As is known in the prior art, an MPEG encoded video signal includes autonomously encoded pictures (I-pictures) and predictively encoded pictures (P-pictures and B-pictures). The selected parts constituting the ancillary signal may be, for example, the I-pictures. In that case, the arrangement is simple because motion compensation circuitry and a large amount of memory can be dispensed with. An embodiment of the arrangement is characterized in that said selected parts are the DC coefficients of autonomously encoded pictures. Such an arrangement is extremely simple.

MPEG also allows parts of the signal to be scrambled, whereas other parts remain unscrambled. A further embodiment of the arrangement is characterized in that the selected parts are the unscrambled parts of the video signal.

The arrangement provides an ancillary video signal having a lower quality than the main signal from which it is derived. Various applications thereof are conceivable. A picture-in-picture television receiver, for example, may comprise the arrangement so as to obtain the ancillary signal for display as the picture-in-picture. In a multi-picture-in-picture television receiver, the arrangement may be used to decode a plurality of elementary video streams, and simultaneously display the respective ancillary signals as a mosaic picture. In a video recorder, the arrangement may be used to obtain a low-quality version of a video signal for separate recording so as to be reproduced at higher playback speeds. If the ancillary signal comprises the unscrambled parts of a scrambled main signal, it allows a video program to be viewed free of charge but at a low quality. The ancillary signal thus acts as an "appetizer", attracting the viewer's attention to the presence and contents of the main signal. When recorded simultaneously with the main signal on a digital video recorder, the ancillary signal also assists the user in finding the beginning of a scrambled program on tape.

The arrangement may also be used in transmitters. According to the invention, a transmitter for transmitting a digital video signal encoded as an MPEG elementary video stream, is characterized in that the transmitter comprises the arrangement for decoding said video signal, and means for transmitting the ancillary video signal as a further elementary video stream. The MPEG standard allows a program to comprise more than one elementary video stream. The ancillary video signal thus transmitted may serve the purposes mentioned before. For decoding the ancillary signal, a simple decoder is adequate. The transmitted ancillary signal requires only a low bitrate.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows a diagram of an arrangement for carrying out the method according to the invention.

FIG. **2** shows a flow chart illustrating the operation of the arrangement shown in FIG. **1**.

FIG. **3** shows a diagram of another embodiment of the arrangement for carrying out the method according to the invention.

FIG. **4** shows a flow chart illustrating the operation of the arrangement shown in FIG. **3**.

FIG. **5** shows a diagram of an arrangement for transmitting the ancillary video signal as an elementary MPEG bitstream of the same program as the main video signal.

FIG. **6** shows a diagram of a digital picture-in-picture television receiver according to the invention.

FIG. **7** shows a diagram of a digital multi-picture-in-picture television receiver according to the invention.

FIGS. **8A** and **8B** show embodiments of a digital video recorder according to the invention.

DESCRIPTION OF EMBODIMENTS

FIG. **1** shows a diagram of an arrangement according to the invention. The arrangement comprises a variable-length decoder **10** (hereinafter VLD), an inverse quantizer **11** and a picture memory **12**. The arrangement receives an elementary video stream representing a main video signal Vm and derives therefrom an ancillary video signal Va. The main video signal Vm is assumed to have been encoded according to "ISO/IEC CD 13818-2: Information technology—Generic coding of moving pictures and associated audio information—Part 2: Video", Dec. 1, 1993, also referred to as the MPEG2 video coding standard. For understanding the invention, it suffices to mention that the main signal Vm

includes autonomously encoded pictures (I-pictures) and predictively encoded pictures (P- and B-pictures). Each picture has been divided into blocks of 8*8 pixels and each block has been transformed to spectral coefficients. The relevant coefficients are subjected to a combination of Huffman coding and runlength coding. Four luminance blocks and associated chrominance blocks constitute a macroblock and a plurality of macroblocks constitute a slice. The first (DC) coefficient of blocks of I-pictures represents the average luminance and chrominance of an 8*8 pixel block. The bitstream Vm further includes overhead information such as syncwords, picture type parameters, and the like.

The operation of the arrangement shown in FIG. **1** will now be elucidated with reference to a flow chart shown in FIG. **2**. In a first step **20**, the VLD reads the input bitstream and discards all data until a picture start code is encountered. Data defining a picture is now being received. In a step **21**, the picture coding type accommodated in the picture header is decoded. In a step **22**, it is established whether said picture coding type indicates that an I-picture is being received. If that is not the case, the VLD returns to step **20** to await the next picture start code. If the picture is an I-picture, the VLD successively awaits the reception of a slice header (step **23**) and the reception of a macroblock (step **24**).

In a step **25**, the VLD decodes and outputs the DC coefficient of a block within the current macroblock. In a step **26**, the subsequent coefficients up to the detection of an end-of-block code are discarded. In a step **27**, it is ascertained whether all blocks of a macroblock have been processed. As long as that is not the case, the VLD returns to step **25**. In a step **28**, it is ascertained whether all macroblocks of a slice have been processed. As long as that is not the case, the VLD returns to step **24**. Finally, it is ascertained in a step **29** whether all slices of the picture have been processed. As long as that is not the case, the VLD returns to step **23**. If all slices have been processed, the VLD returns to step **20** in order to search the next I-picture in the bitstream.

The VLD thus extracts the DC coefficients of I-pictures from the input bitstream. As shown in FIG. **1**, said coefficients are supplied to the inverse quantizer **11** and then stored in memory **12**. Each DC coefficient represents the average luminance and chrominance value of an 8*8 pixel block of the main video I-pictures. The ancillary video signal is obtained by reading out said memory with an appropriate time basis.

In an alternative embodiment, steps **25** and **26** are modified so as to decode all coefficients of a block. In that case, the ancillary signal comprises I-pictures and is a temporally reduced version of the main signal.

FIG. **3** shows a diagram of another embodiment of the arrangement for carrying out the method according to the invention. In this arrangement the bitstream representing the main video signal Vm is supplied to a memory **30** and variable-length decoder **31**. The variable-length decoder analyses the bitstream and generates a write enable signal WE so as to determine which part of the bitstream is stored in the memory. The memory is read out at a lower bitrate so as to constitute an elementary video stream representing the ancillary video signal Va.

The operation of the arrangement shown in FIG. **3** will now be elucidated with reference to a flow chart shown in FIG. **4**. The steps **20–22** are the same as the corresponding steps shown in FIG. **2**. Thus, in the step **20**, the VLD reads the input bitstream and discards all data until a picture start code is encountered. In the step **21**, the picture coding type accommodated in the picture header is decoded. In the step **22**, it is ascertained whether said picture coding type indicates that an I-picture is being received. If that is not the case, the VLD returns to step **20** to await the next picture start code.

If the picture is an I-picture, a step **40** is performed in which the VLD allows the picture header to be kept in the memory by generating an appropriate write enable signal. In a step **41**, the slice header is received and stored in the memory. A macroblock is now being received. In a step **42**, all macroblock data up to the first block is written in the memory.

In a step **43**, the VLD detects the presence of a DC coefficient of a block within the current macroblock and allows this coefficient to be stored in the memory. In a step **44**, the subsequent coefficients of the block up to the detection of an end-of-block code are discarded. The VLD refrains from generating the write enable signal while said coefficients are being received. In a step **45**, the end-of-block code is stored in the memory. Steps **27–29** are the same as the corresponding steps shown in FIG. **2**. They facilitate the check of whether or not the current I-picture has been processed.

The arrangement shown in FIG. **3** thus copies the main bitstream Vm in memory **30**, thereby ignoring the P- and B-pictures as well as the non-DC-coefficients of I-pictures. The ancillary video signal Va obtained by reading out the memory **30** is the same as that created by the arrangement shown in FIG. **1** but is now suitable for transmission as a further elementary signal. It is a low bitrate replica of the main signal, with a reduced spatial and temporal resolution.

FIG. **5** shows a diagram of a transmitter according to the invention. The transmitter comprises a demultiplexer **50**, a transcoder **51**, a circuit **52** for regenerating program specific information, and a remultiplexer **53**. The arrangement receives a packetized transport stream TS1. Said transport stream comprises a plurality of audiovisual programs, each program being formed by one or more elementary streams (e.g. video, audio, data). The transport stream also comprises packets accommodating program-specific information (PSI). PSI packets specify which programs are available, as well as how many and which elementary streams each program comprises. A detailed description of transport streams and program-specific information can be found in "ISO/IEC CD 13818-1: Information technology—Generic coding of moving pictures and associated audio information—Part 1: Systems", Dec. 1, 1993, also known as the MPEG2 systems standard.

Demultiplexer **50** selects an elementary video stream Vm from which an ancillary signal Va is to be derived. Other elementary streams E1,E2,E3 remain unprocessed in this embodiment. The main video signal Vm is applied to transcoder **51** which may take the form of the arrangement shown in FIG. **3**, already discussed. The transcoder outputs an ancillary video signal Va in the form of a further elementary stream. Demultiplexer **50** also extracts program-specific information packets PSI1 from the transport stream TS1 and applies them to circuit **52**. This circuit updates the program-specific information so as to specify that ancillary signal Va is present in output transport stream TS2. The circuit further specifies that Va is associated with the same audiovisual program as the main elementary stream Vm from which it has been derived. The updated program-specific information PSI2 and the ancillary elementary stream Va are then added by remultiplexer **53** to the original elementary streams and retransmitted as a new transport stream TS2.

FIG. 6 shows a diagram of a digital picture-in-picture (PIP) television receiver according to the invention. The receiver comprises a demultiplexer 60, an MPEG2 audio decoder 61, an MPEG2 video decoder 62, a PIP-decoder 63, and a video mixer 64. The demultiplexer 60 receives an MPEG2 transport stream TS and extracts therefrom an elementary audio stream A1 and an elementary video stream V1 associated with a desired program. The elementary streams A1 and V1 are decoded by audio decoder 61 and video decoder 62, respectively. The decoded audio signal is applied to a loudspeaker 65. The demultiplexer further extracts from the transport stream TS a further elementary video stream V2 associated with a different program which is to be displayed as picture-in-picture. The further elementary stream V2 is decoded by PIP-decoder 63 and converted into a signal V2' with a reduced size and temporal resolution. Both video signals V1 and V2' are mixed in video mixer 64 and displayed on a display screen 66.

In a first embodiment of the PIP-receiver, elementary stream V2 defines a full-size, full-resolution MPEG-encoded video signal, including I, P and B-pictures. In this embodiment, PIP-decoder 63 takes the form of the circuit shown in FIG. 1, already discussed. In a second embodiment of the PIP-receiver, the elementary stream V2 is assumed to be an ancillary video stream transmitted by an arrangement as shown in FIG. 5. As explained with reference to FIG. 5, the elementary stream V2 comprises DC-coefficients of I-pictures only. In this embodiment, PIP-decoder 63 also takes the form of the circuit shown in FIG. 1, already discussed. However, the variable-length decoder (10 in FIG. 1) is simpler because various types of overhead data are absent in the bitstream and thus do not need to be processed.

FIG. 7 shows a diagram of a digital multi-picture-in-picture (MPIP) television receiver according to the invention. The receiver comprises a transport stream demultiplexer 70, an MPEG2 audio decoder 71, a loudspeaker 72, an MPEG2 video decoder 73, a MPIP-decoder 74, a display screen 75 and a control circuit 76. The demultiplexer 70 receives an MPEG2 transport stream TS and extracts therefrom an elementary audio stream Aj and an elementary video stream Vj, both associated with a program number j. The elementary streams Aj and Vj are decoded by audio decoder 71 and video decoder 73, respectively. The decoded audio signal is applied to loudspeaker 72. The decoded video signal is displayable, via a switch 77, on display screen 75. The demultiplexer further extracts from the transport stream TS a further elementary video stream Vi associated with a program i. The further elementary stream Vi may define a full-size, full-resolution MPEG-encoded video signal, including I, P and B-pictures. The further elementary video stream Vi may also be an ancillary video stream transmitted by an arrangement as shown in FIG. 5. In the latter case, the ancillary video signal comprises, as explained above, DC-coefficients of I-pictures only.

MPIP-decoder 74 is adapted to decode the ancillary signal Vi. The decoder comprises a variable-length decoder 741, an inverse quantizer 742 and a memory 743. The decoder has the same structure as the arrangement shown in FIG. 1. However, memory 743 now has a plurality of memory sections, addressed by a write address WA, each section having the capacity to store a respective small-size picture.

In operation, control circuit 76 receives from demultiplexer 70 the transport packets accommodating program-specific information PSI. As already mentioned before, said packets specify which programs are available, as well as how many and which elementary streams each program comprises. The control circuit is adapted to read from the PSI-data, for each available program i, the packet identifier PID defining the transport packets conveying the ancillary video signal Vi associated therewith. For a plurality of different programs, the relevant PIDs are successively applied to the demultiplexer 70 so as to apply the associated ancillary video signal Vi to MPIP-decoder 74. Each decoded small-size picture is stored in a section of memory 743 under the control of the write address WA generated by control circuit 76. The plurality of small-size pictures together constitutes a mosaic video picture which can be displayed under user control on display screen 75 via the switch 77.

Upon user-selection of one of the displayed miniature pictures (e.g. by a cursor device not shown), the control circuit 76 converts the selected display screen position into the program number associated therewith, and controls demultiplexer 70 so as to select the audio stream Aj and video stream Vj associated with the selected program. The control circuit further controls switch 77 so as to display the selected program in full size and resolution on display screen 75 and to reproduce its sound via loudspeaker 72.

FIGS. 8A and 8B show two embodiments of a digital video recorder according to the invention. The recorder receives an MPEG2 transport stream TS and comprises a demultiplexer 80 to select therefrom an elementary audio stream A and an elementary video stream V. The elementary video stream is assumed to have been scrambled such that only the predictively encoded (P and B) pictures are scrambled. Both elementary streams are recorded on a digital storage medium 81.

In the embodiment shown in FIG. 8A, the video stream V is further applied to a transcoder 82 which may take the form as shown in FIG. 1. In the embodiment shown in FIG. 8B, the demultiplexer further selects an ancillary elementary stream Va, which is transmitted by an arrangement as shown in FIG. 5. The ancillary stream is applied to a "simple" MPEG decoder 89 as explained hereinbefore with reference to FIG. 6.

In both embodiments, the ancillary signal Va comprises the DC-coefficients of I-pictures of the same program as video signal V. This ancillary signal is separately recorded on storage medium 81. Upon normal playback, the recorded audio and video elementary streams A and V are decoded by MPEG2 audio decoder 83 and MPEG2 video decoder 85, respectively. The audio signal is applied to an audio output terminal 84. If the video signal has been scrambled, it can only be displayed when processed by a descrambler 86. The descrambler may take the form of a circuit which is activated only upon insertion of a smart card holding a sufficient amount of credit. These types of descramblers are known per se in the art. The decoded and descrambled video signal is then applied, via a switch 87, to a video output 88.

The video recorder can optionally output, via switch 87, the separately recorded ancillary signal Va. Said signal comprises DC-oefficients of I-pictures only and is thus not scrambled. Displaying this signal in reduced size or, after suitable upsampling (not shown), in full size but with reduced resolution, allows the user to scan the storage medium 81 for locating the start of a particular scrambled program without having yet to pay therefor. It is to be noted that in the embodiment of FIG. 8B the "simple" decoder 89 can also be located in the reproduction part (i.e. between storage medium 81 and switch 87) of the video recorder.

In summary, the invention relates to a method and arrangement for deriving an ancillary signal from a compressed digital video signal (e.g. MPEG). The DC coefficients of autonomously encoded pictures (I-pictures) are

selected from the compressed signal. The ancillary signal thus obtained can be used for display in a (multi-) picture-in-picture television receiver. If the main signal is scrambled, the ancillary signal can be used as an "appetizer" in order to encourage the user to pay a subscription fee. The ancillary signal can separately be recorded in digital video recorders so as to assist the user in finding the beginning of a scrambled program on tape. The ancillary signal can also be generated at the transmitter end and transmitted at a low bit rate. A decoder for decoding such an ancillary signal is considerably. simpler and less expensive than a full-spec MPEG decoder.

What is claimed is:

1. A transmitter comprising:

means for generating a main elementary video stream that includes autonomously encoded pictures and predictively encoded pictures;

means for selecting from the main elementary video stream only the autonomously encoded pictures;

means for arranging the autonomously encoded pictures to generate an ancillary elementary video stream;

means for multiplexing the main elementary video stream and the ancillary elementary video stream to generate a transport stream; and,

means for transmitting the transport stream wherein the transport stream is transmitted over a transmission channel to a receiver that is located remotely from the transmitter.

2. The transmitter as set forth in claim 1, wherein the transport stream further includes a plurality of additional elementary video streams corresponding to a plurality of different programs.

3. The transmitter as set forth in claim 1, wherein the ancillary elementary video stream comprises a low bitrate replica of the main elementary video stream.

4. The transmitter as set forth in claim 1, wherein:

the main elementary video stream comprises a main MPEG-encoded elementary video stream;

the ancillary elementary video stream comprises an ancillary MPEG-encoded elementary video stream; and,

the transport stream comprises an MPEG transport stream.

5. The transmitter as set forth in claim 1, wherein the means for selecting comprises means for selecting from the main elementary video stream all of the autonomously-encoded pictures included in the main elementary video stream, but none of the predictively-encoded pictures included in the main elementary video stream.

6. The transmitter as set forth in claim 3, wherein the low bitrate replica of the main elementary video stream is arranged to be displayed as a picture-in-picture by the receiver.

7. The transmitter as set forth in claim 1, wherein the ancillary elementary video stream is arranged to be displayed by the decoder as a separate one of a plurality of pictures that together form a mosaic picture.

8. A receiver, comprising:

means for receiving a transport stream that includes multiplexed main and ancillary elementary video streams;

means for de-multiplexing the main and ancillary elementary video streams to generate separate main and ancillary elementary video streams; and,

wherein the main elementary video stream includes autonomously encoded pictures and predictively encoded pictures, and the ancillary elementary video stream includes only autonomously encoded pictures from the main elementary video stream.

9. The receiver as set forth in claim 8, wherein the transport stream further includes a plurality of additional elementary video streams corresponding to a plurality of different programs.

10. The receiver as set forth in claim 8, wherein the transport stream is received from a transmitter that is located remotely from the receiver.

11. The receiver as set forth in claim 8, wherein:

the main elementary video stream comprises a main MPEG-encoded elementary video stream;

the ancillary elementary video stream comprises an ancillary MPEG-encoded elementary video stream; and,

the transport stream comprises an MPEG transport stream.

12. The receiver as set forth in claim 8, wherein the ancillary elementary video stream contains all of the autonomously-encoded pictures included in the main elementary video stream, but none of the predictively-encoded pictures included in the main elementary video stream.

13. The receiver as set forth in claim 8, further comprising means for displaying the ancillary elementary video stream as a picture-in-picture.

14. The receiver as set forth in claim 8, further comprising means for displaying the ancillary elementary video stream as a separate one of a plurality of pictures that together form a mosaic picture.

15. A system, comprising:

a transmitter that includes:

means for generating a main elementary video stream that includes autonomously encoded pictures and predictively encoded pictures;

means for selecting from the main elementary video stream only the autonomously encoded pictures;

means for arranging the autonomously encoded pictures to generate an ancillary elementary video stream;

means for multiplexing the main elementary video stream and the ancillary elementary video stream to generate a transport stream; and,

means for transmitting the transport stream; and,

a receiver that includes:

means for receiving the transport stream; and,

means for de-multiplexing the main and ancillary elementary video streams to generate separate main and ancillary elementary video streams.

16. The system as set forth in claim 15, wherein the transport stream further includes a plurality of additional elementary video streams corresponding to a plurality of different programs.

17. The system as set forth in claim 15, wherein:

the main elementary video stream comprises a main MPEG-encoded elementary video stream;

the ancillary elementary video stream comprises an ancillary MPEG-encoded elementary video stream; and,

the transport stream comprises an MPEG transport stream.

18. The system as set forth in claim 15, wherein the ancillary elementary video stream contains all of the autonomously-encoded pictures included in the main elementary video stream, but none of the predictively-encoded pictures included in the main elementary video stream.

19. A method, comprising:

generating a main elementary video stream that includes autonomously encoded pictures and predictively encoded pictures;

selecting from the main elementary video stream only the autonomously encoded pictures;

arranging the autonomously encoded pictures to generate an ancillary elementary video stream;

multiplexing the main elementary video stream and the ancillary elementary video stream to generate a transport stream; and,

transmitting the transport stream over a transmission channel to a receiver that is located remotely from the transmitter.

20. The transmitter as set forth in claim 1, further comprising means for updating and transmitting a program specific information stream to specify the presence of the ancillary elementary video stream and the associated main elementary video stream.

21. The receiver as set forth in claim 8, wherein the means for de-multiplexing have been arranged to obtain an updated program-specific information stream that specifies the presence of the ancillary elementary video stream and the associated elementary video stream.

22. The system as set forth in claim 15, wherein the transport stream further includes an updated program-specific information stream that specifies the presence of the ancillary elementary video stream and the associated MPEG elementary video stream.

* * * * *

US006671322B2

(12) **United States Patent**
Vetro et al.

(10) **Patent No.: US 6,671,322 B2**
(45) **Date of Patent: Dec. 30, 2003**

(54) **VIDEO TRANSCODER WITH SPATIAL RESOLUTION REDUCTION**

(75) Inventors: **Anthony Vetro**, Staten Island, NY (US); **Huifang Sun**, Cranbury, NJ (US); **Peng Yin**, Princeton, NJ (US); **Bede Liu**, Princeton, NJ (US); **Tommy C. Poon**, Murray Hill, NJ (US)

(73) Assignee: **Mitsubishi Electric Research Laboratories, Inc.**, Cambridge, MA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 245 days.

(21) Appl. No.: **09/853,394**

(22) Filed: **May 11, 2001**

(65) **Prior Publication Data**

US 2003/0016751 A1 Jan. 23, 2003

(51) **Int. Cl.**[7] **H04N 7/12**

(52) **U.S. Cl.** **375/240.16**; 375/240.13; 375/240.21

(58) **Field of Search** 375/240.25, 240.03, 375/240.12, 240.16, 240.21, 240.01; 348/410.1, 412.1, 413.1, 416.1, 420.1; 382/232, 236, 239

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,870,146	A *	2/1999	Zhu	375/240.03
6,275,536	B1 *	8/2001	Chen et al.	375/240.25
6,310,915	B1 *	10/2001	Wells et al.	375/240.03
6,404,814	B1 *	6/2002	Apostolopoulos et al.	375/240.12
6,434,196	B1 *	8/2002	Sethuraman et al.	375/240.12
6,441,754	B1 *	8/2002	Wang et al.	341/50
6,480,670	B1 *	11/2002	Hatano et al.	386/109
6,498,814	B1 *	12/2002	Morel	375/240.12

OTHER PUBLICATIONS

Assuncao et al.; "A Frequency Domain Video Transcoder for Dynamic Bit-Rate Reduction of MPEG-2 Bit Streams"; IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, No. 8, pp. 953–967, 1998.

Panusopone et al.; "Video Format Conversion and Transcoding from MPEG-2 to MPEG-4".

Shanableh et al.; "Heterogeneous Video Transcoding to Lower Spatio-Temporal Resolutions and Different Encoding Formats"; IEEE Transactions on Multimedia, vol. 2, No. 2, pp. 101–110, 2000.

Stuhlmuller et al.; "Analysis of Video Transmission over Lossy Channels"; IEEE Journal on Selected Areas in Communications, vol. 18, No. 6, pp. 1012–1032, 2000.

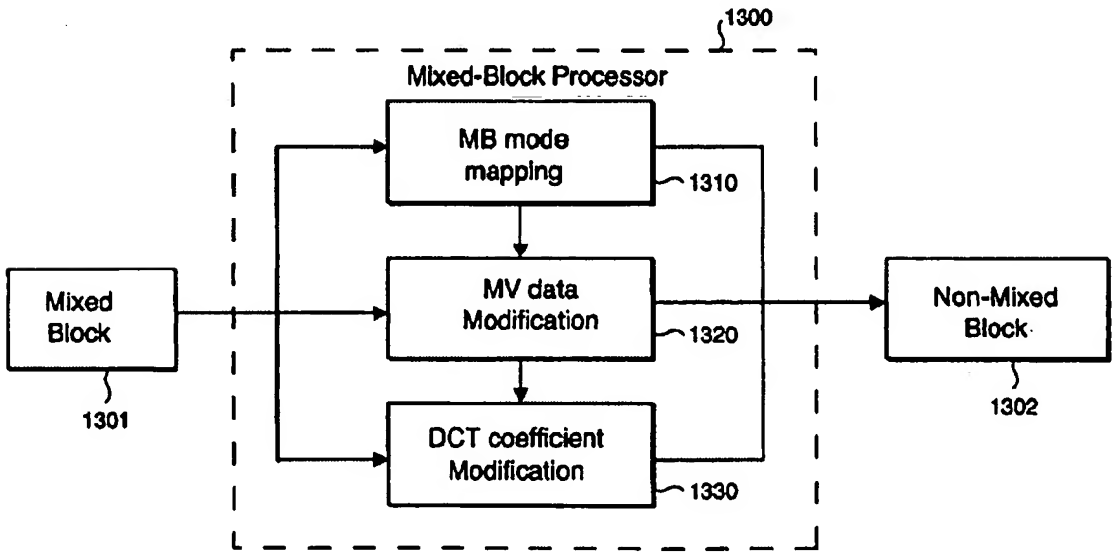
* cited by examiner

Primary Examiner—Vu Le
Assistant Examiner—Behrooz Senfi
(74) *Attorney, Agent, or Firm*—Dirk Brinkman; Andrew Curtin

(57) **ABSTRACT**

A method transcodes groups of macroblocks of a partially decoded input bitstream. The groups of macroblocks include intra-mode and inter-mode macroblocks. Each macroblock includes DCT coefficients, and at least one motion vector. The modes of each group of macroblocks are mapped to be identical only if there is an inter-mode block and an intra-mode macroblock in the group. If any of the macroblocks in the group are mapped, then the DCT coefficients and the motion vector for such mapped macroblocks are modified in accordance with the mapping to generate reduced-resolution macroblock for an output compressed bitstream to compensate for drift.

18 Claims, 19 Drawing Sheets



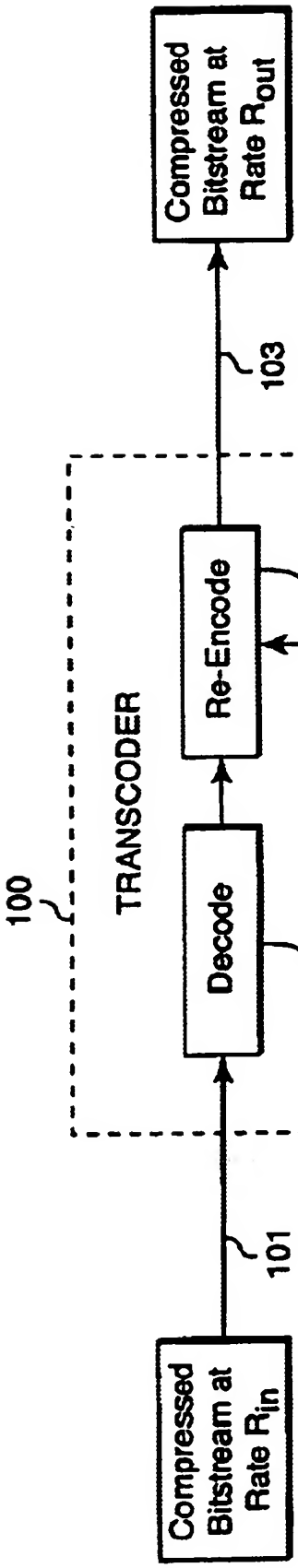


FIG. 1
PRIOR ART

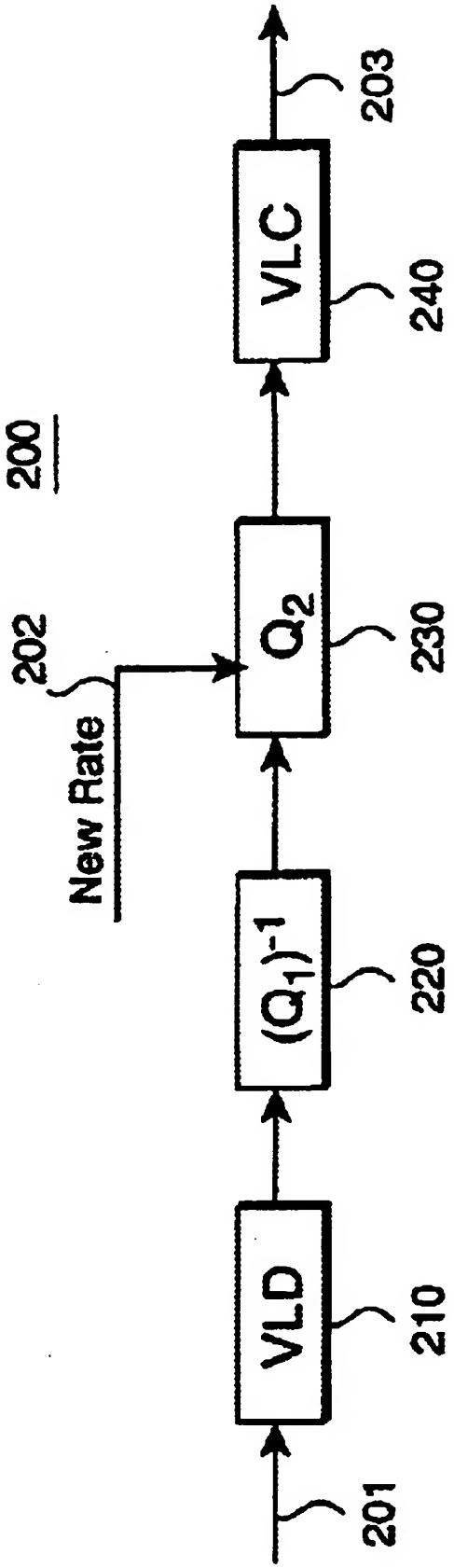


FIG. 2
PRIOR ART

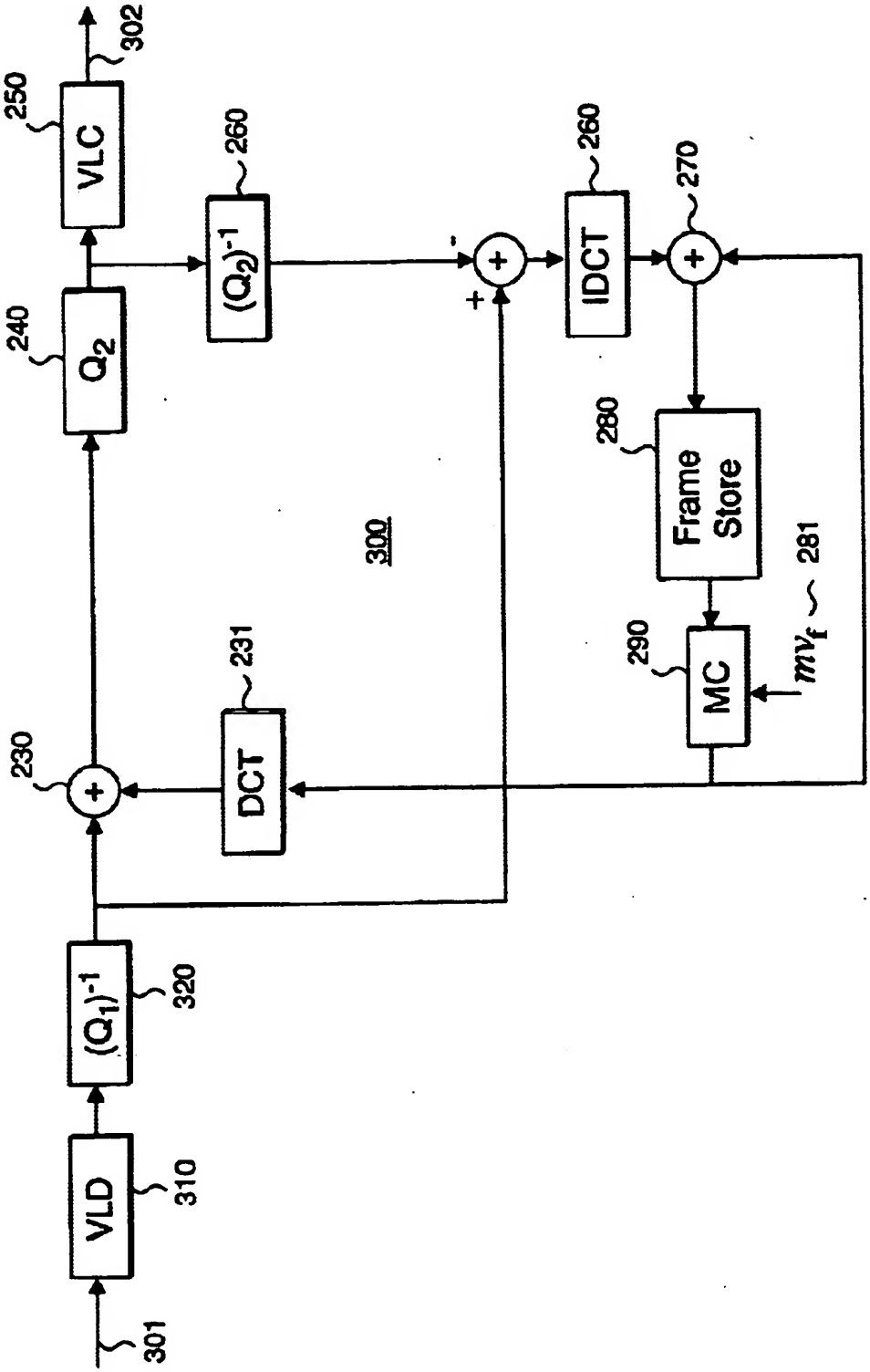


FIG 3
PRIOR ART

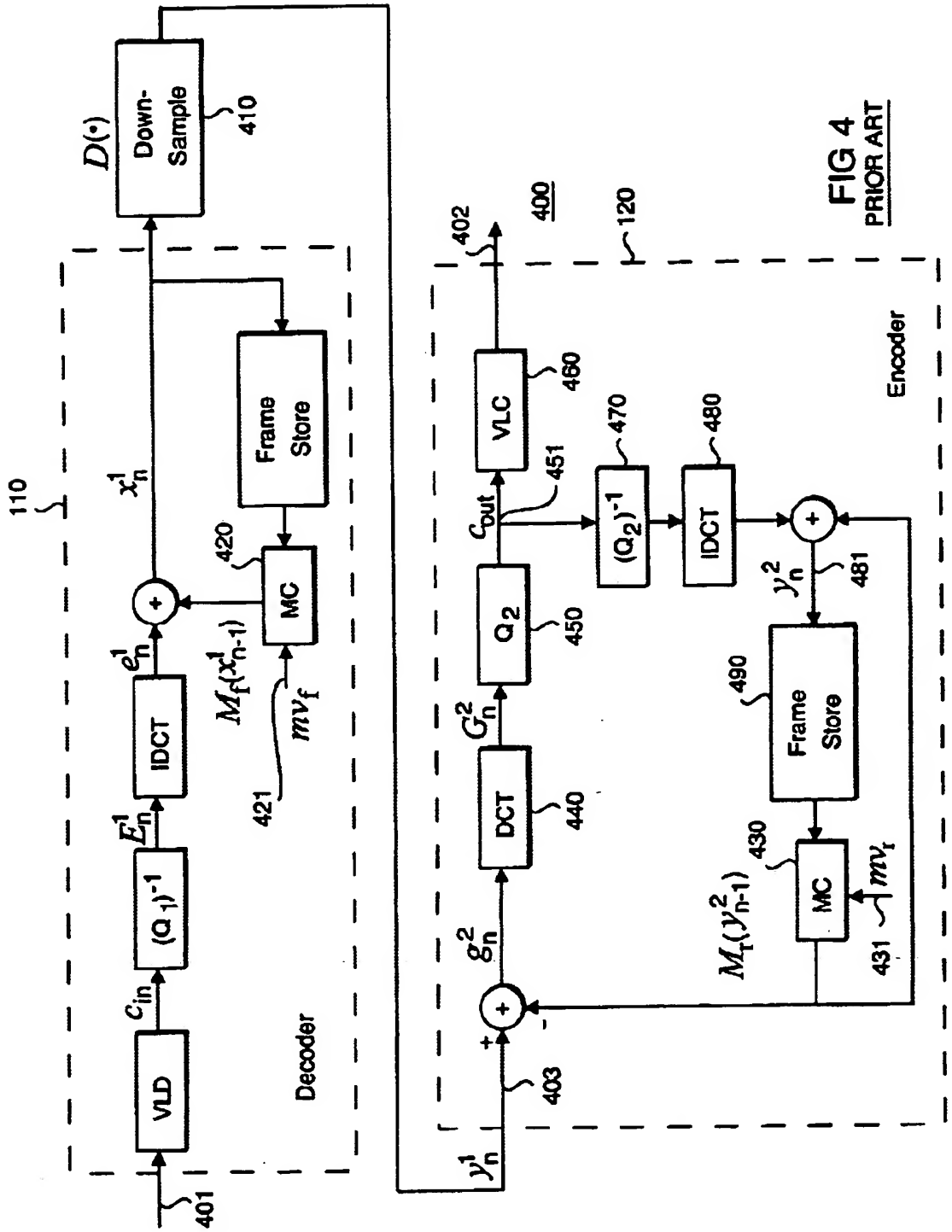


FIG 4
PRIOR ART

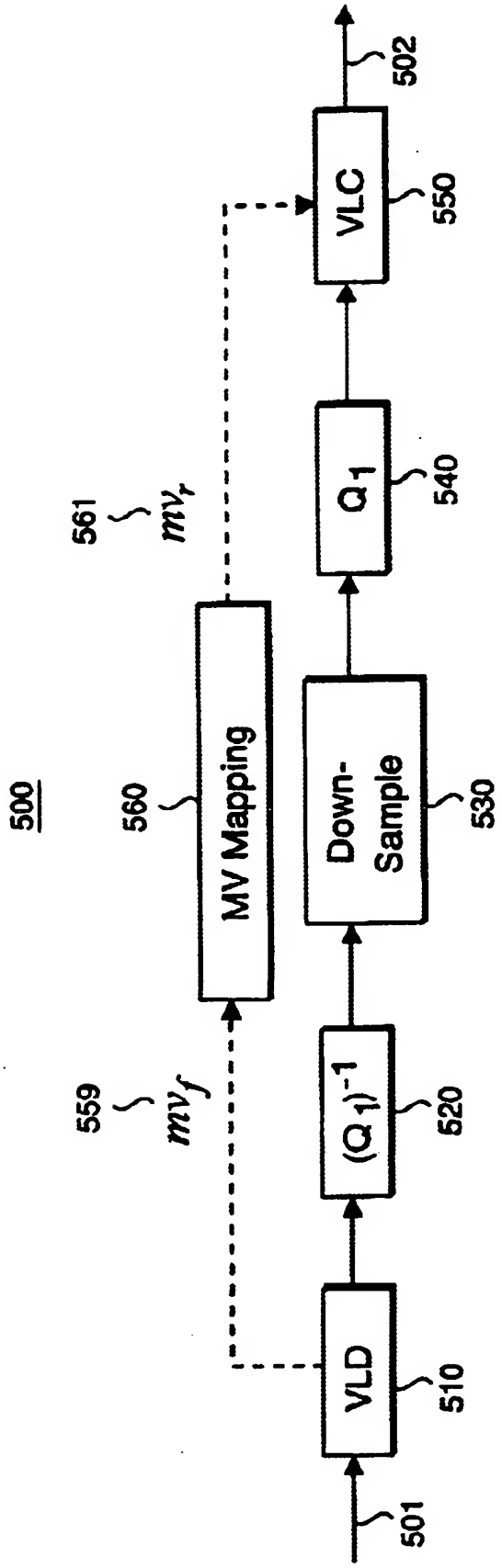


FIG 5
PRIOR ART

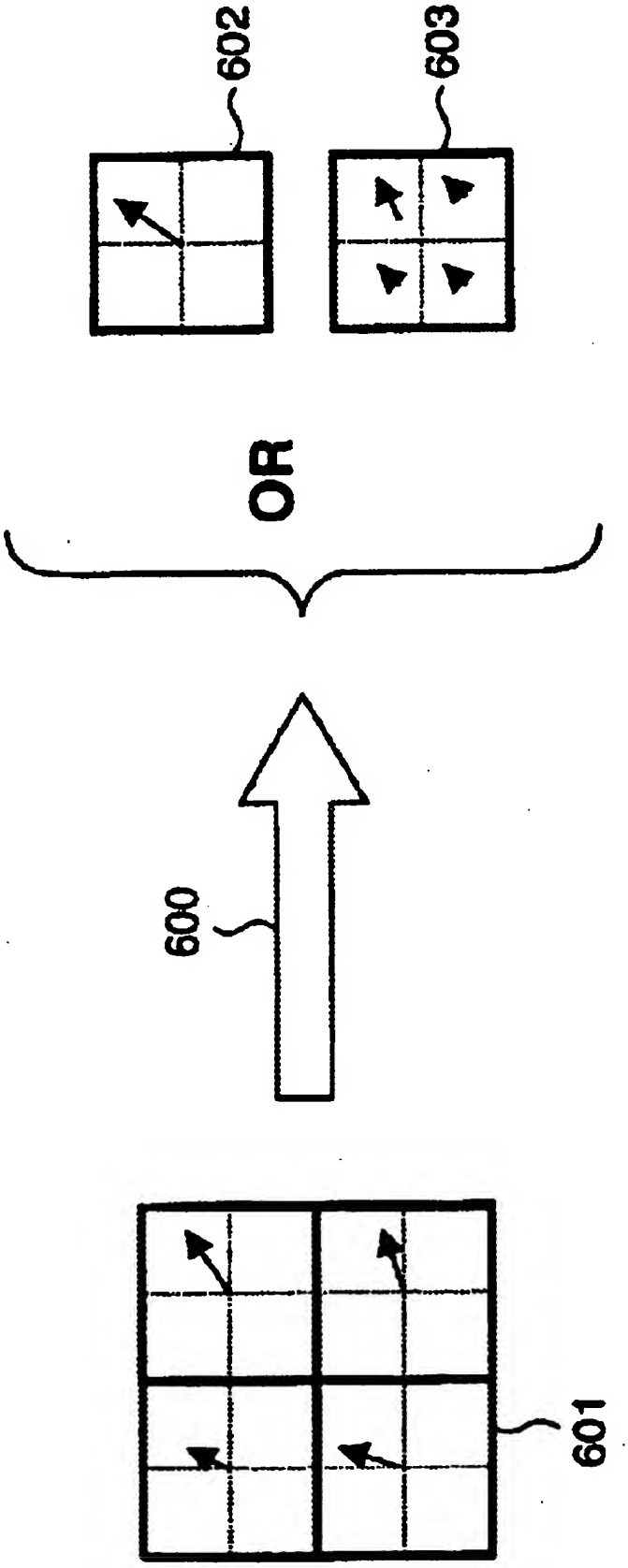


FIG. 6
PRIOR ART

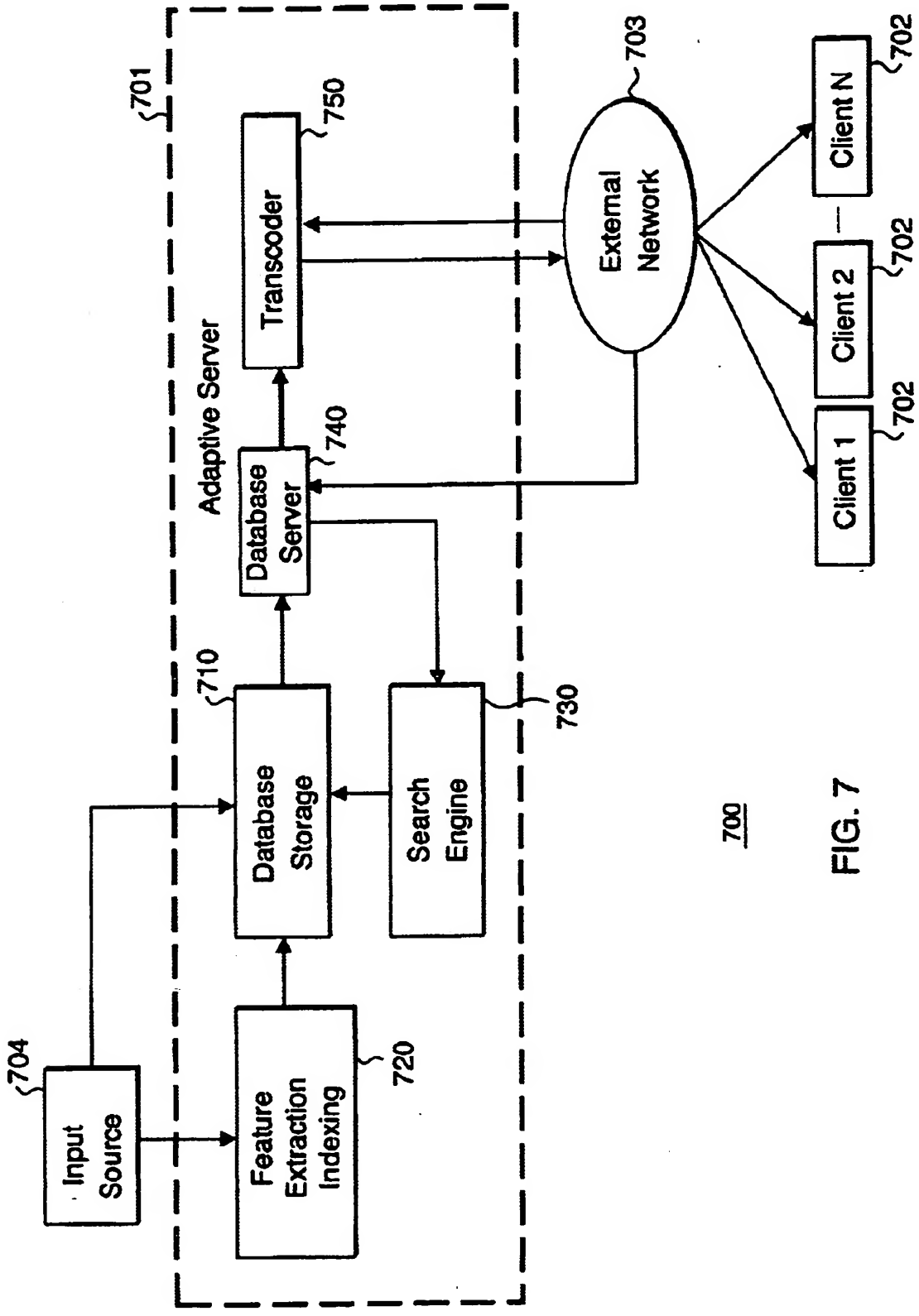

701
Adaptive Server
750
Transcoder
740
Database Server
710
Database Storage
730
Search Engine
720
Feature Extraction Indexing
704
Input Source
703
External Network
702
Client N
702
Client 2
702
Client 1
700

FIG. 7

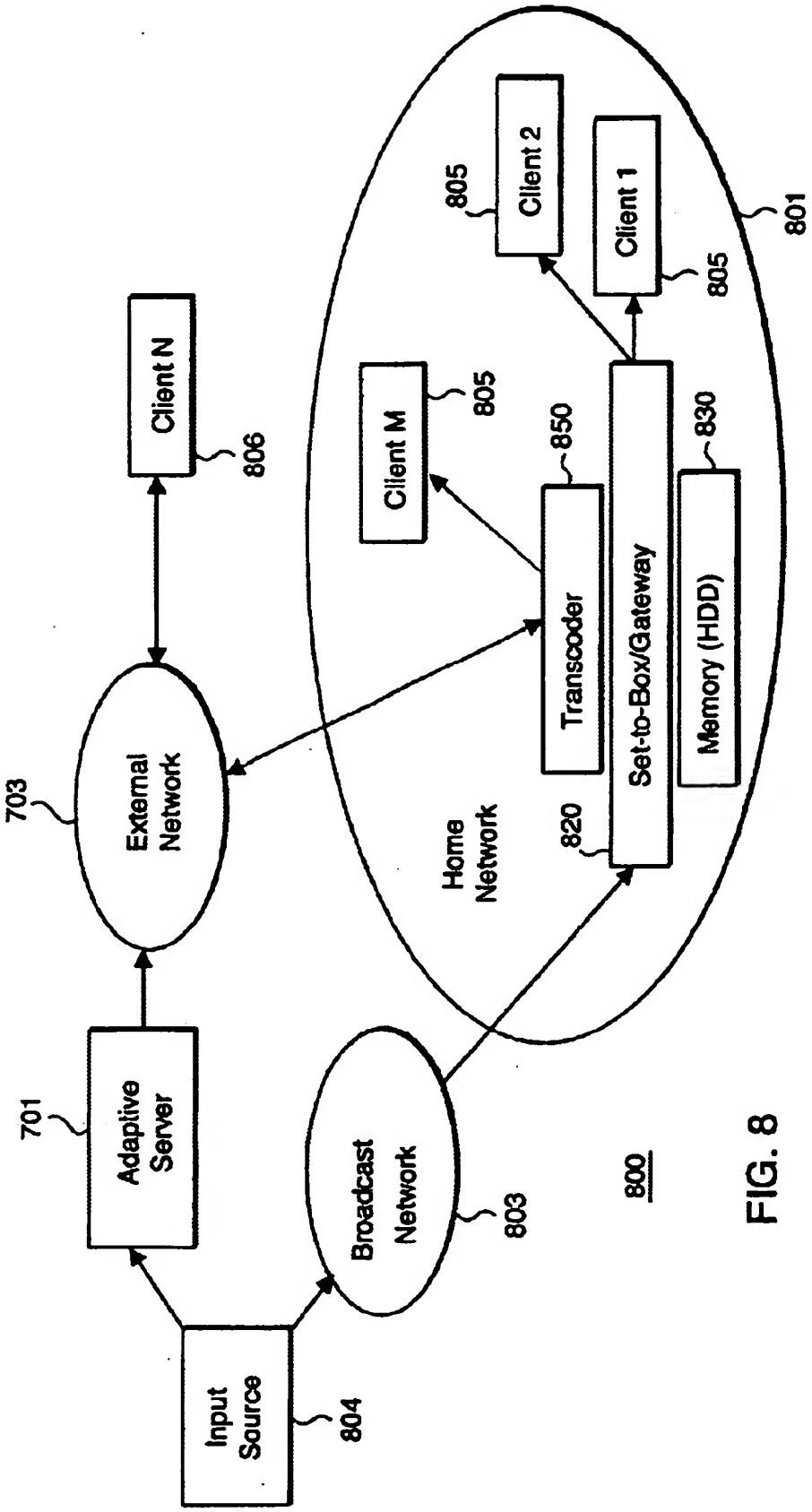

Client N
806
External Network
703
Adaptive Server
701
Input Source
804
Broadcast Network
803
Client 2
805
Client 1
805
Client M
805
801
850
Transcoder
830
Set-to-Box/Gateway
Memory (HDD)
820
Home Network
800

FIG. 8

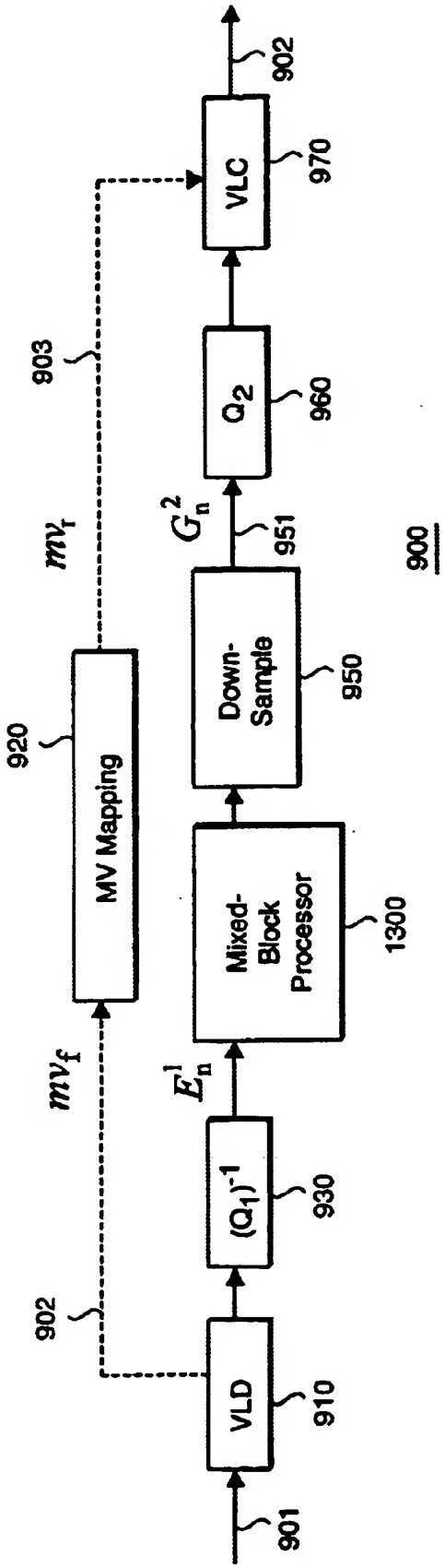

902
VLC
970
903
Q_2
960
mv_f
G_n^2
951
900
Down-Sample
950
920
MV Mapping
Mixed-Block Processor
1300
mv_f
E_n^1
$(Q_1)^{-1}$
930
902
VLD
910
901

FIG. 9

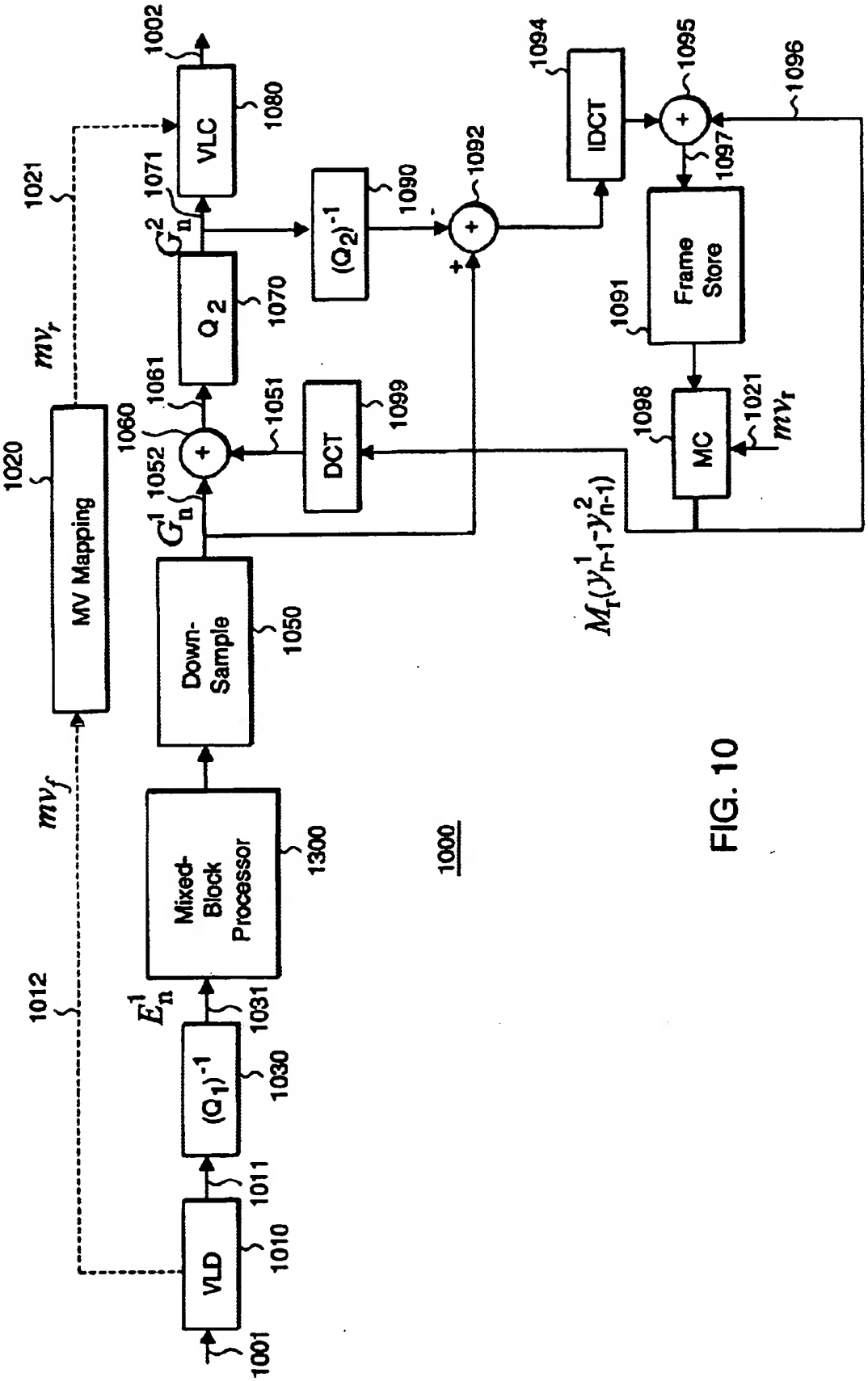

1001
VLD
1010
1011
$(Q_1)^{-1}$
1030
1031
E_n^1
1012
Mixed-Block Processor
1300
mv_f
Down-Sample
1050
MV Mapping
1020
mv_r
1021
G_n^1
1052
1060
1061
Q 2
1070
G_n^2
1071
VLC
1080
1002
1051
DCT
1099
$(Q_2)^{-1}$
1090
1092
IDCT
1094
1095
1096
1097
Frame Store
1091
MC
1098
mv_r
1021
$M_r(y_{n-1}^1-y_{n-1}^2)$
1000

FIG. 10

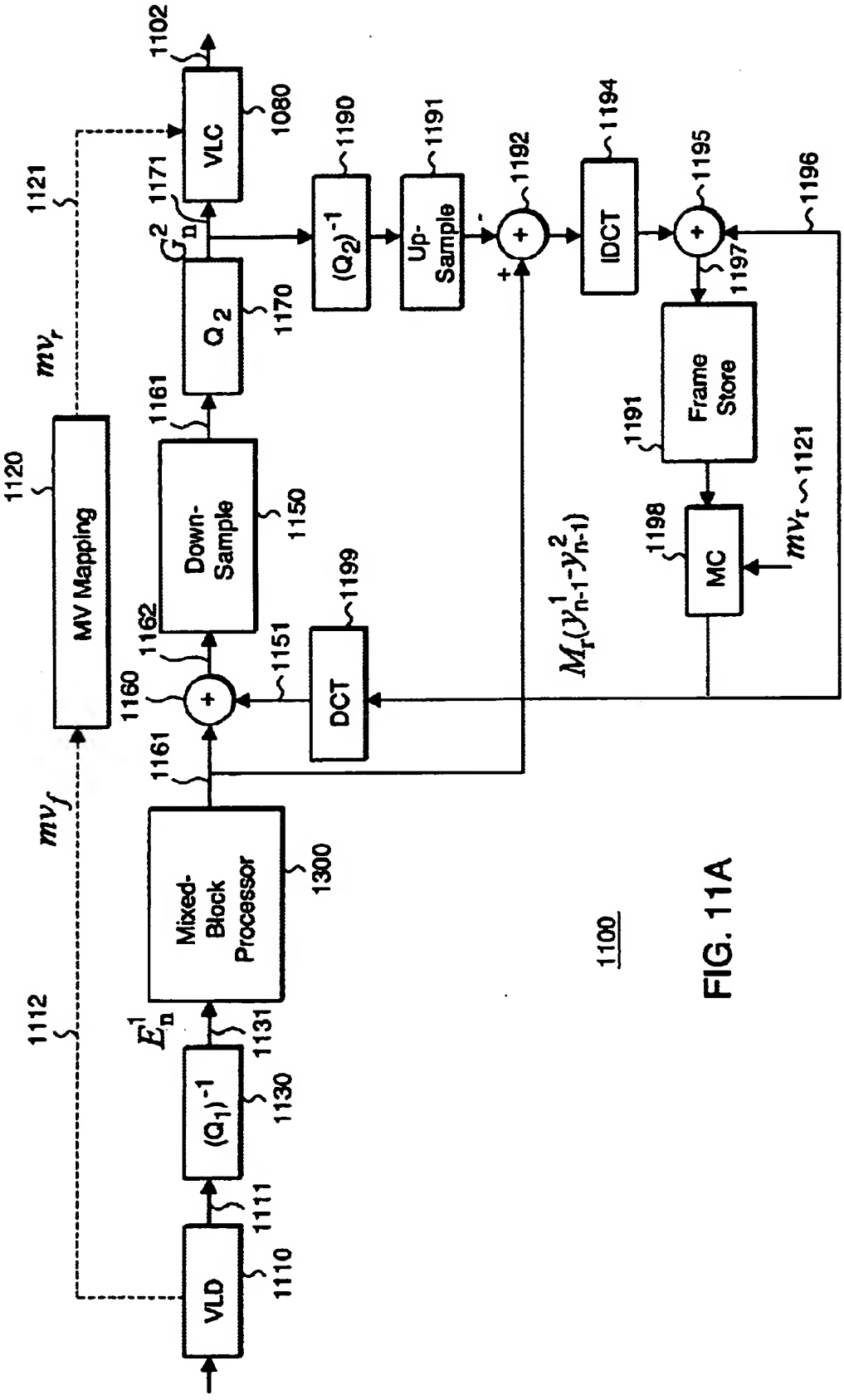

VLD
VLC
MV Mapping
Down-Sample
Up-Sample
Frame Store
MC
DCT
IDCT
Mixed-Block Processor
$(Q_1)^{-1}$
$(Q_2)^{-1}$
Q_2
mv_f
mv_r
E_n^1
G_n^2
$M_r(y_{n-1}^1 - y_{n-1}^2)$
1100
1102
1080
1190
1191
1192
1194
1195
1196
1197
1121
1120
1161
1170
1150
1162
1199
1151
1160
1191
1198
1111
1110
1130
1131
1112
1300

FIG. 11A

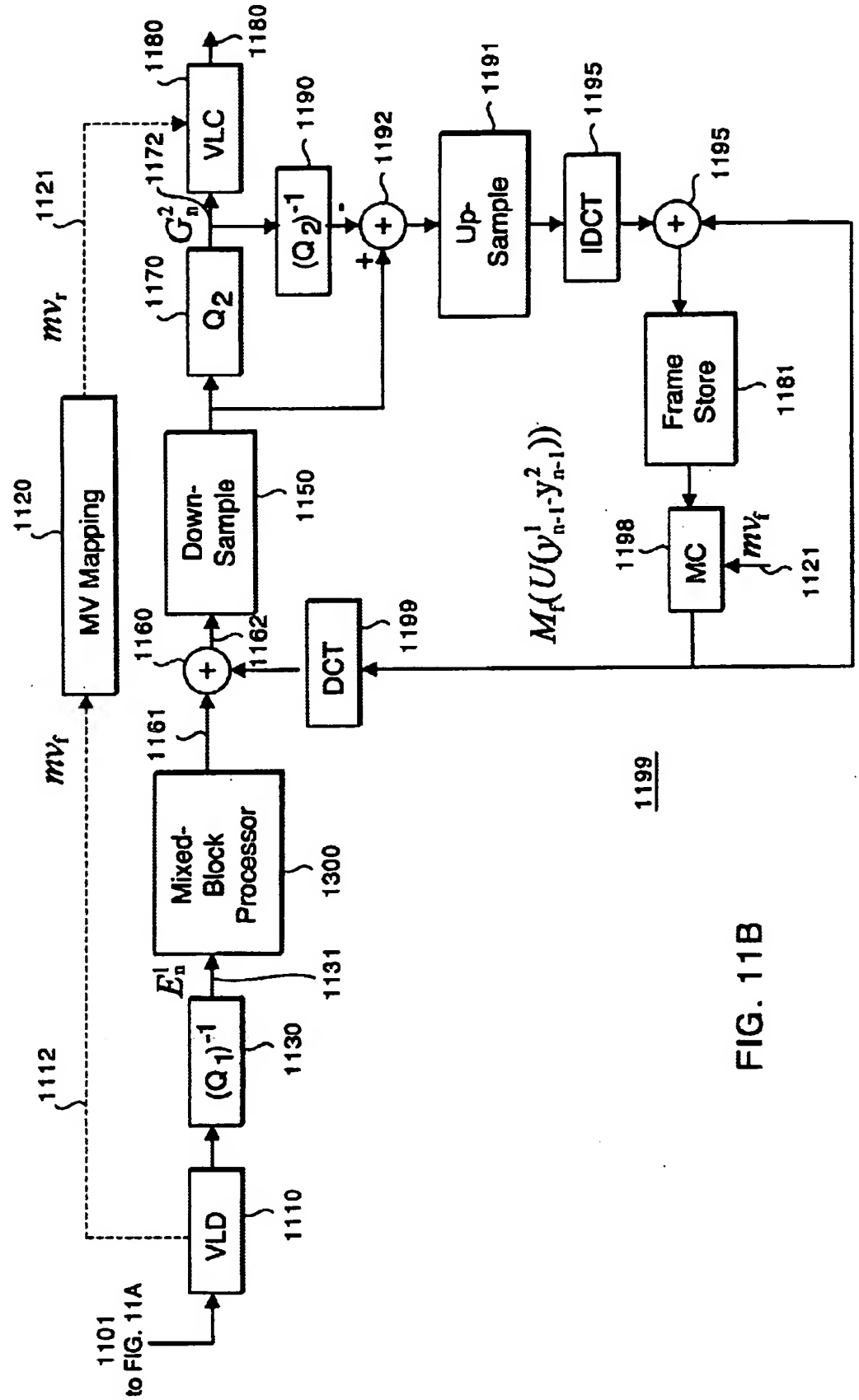

1101
to FIG. 11A
VLD
1110
1112
$(Q_1)^{-1}$
1130
E_n^1
1131
Mixed-Block Processor
1300
1161
mv_f
MV Mapping
1120
mv_r
1121
1160
1162
Down-Sample
1150
1170
Q_2
G_n^2
1172
VLC
1180
1180
DCT
1199
1199
$M_f(U(y_{n-1}^1 - y_{n-1}^2))$
$(Q_2)^{-1}$
1190
1192
Up-Sample
1191
IDCT
1195
1195
Frame Store
1181
MC
1198
mv_f
1121

FIG. 11B

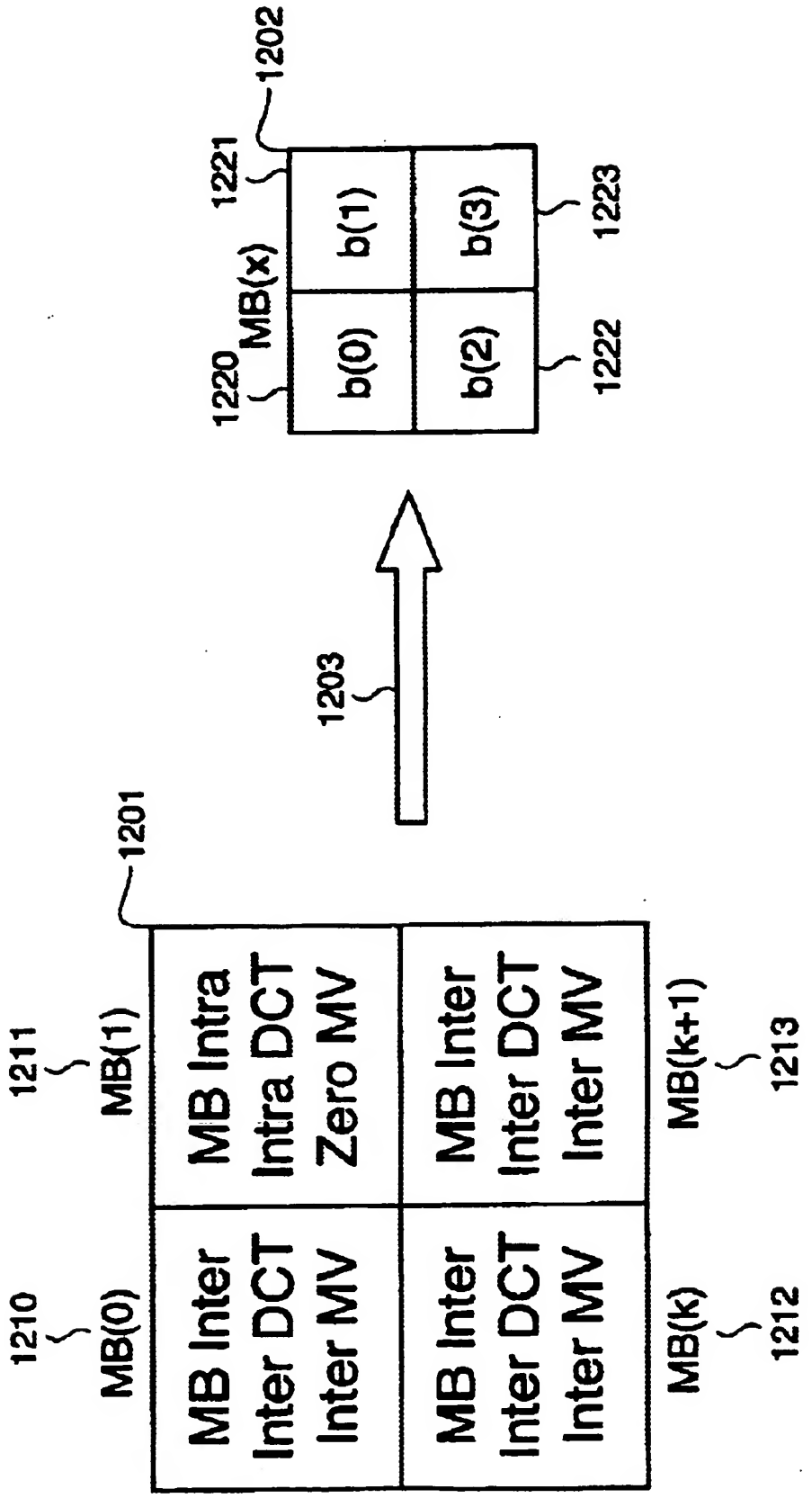

1202
1221
1223
b(1)
b(3)
MB(x)
1220
b(0)
b(2)
1222
1203
1201
1211
MB(1)
MB Intra
Intra DCT
Zero MV
MB Inter
Inter DCT
Inter MV
1210
MB(0)
MB Inter
Inter DCT
Inter MV
MB Inter
Inter DCT
Inter MV
MB(k)
1212
MB(k+1)
1213

FIG. 12

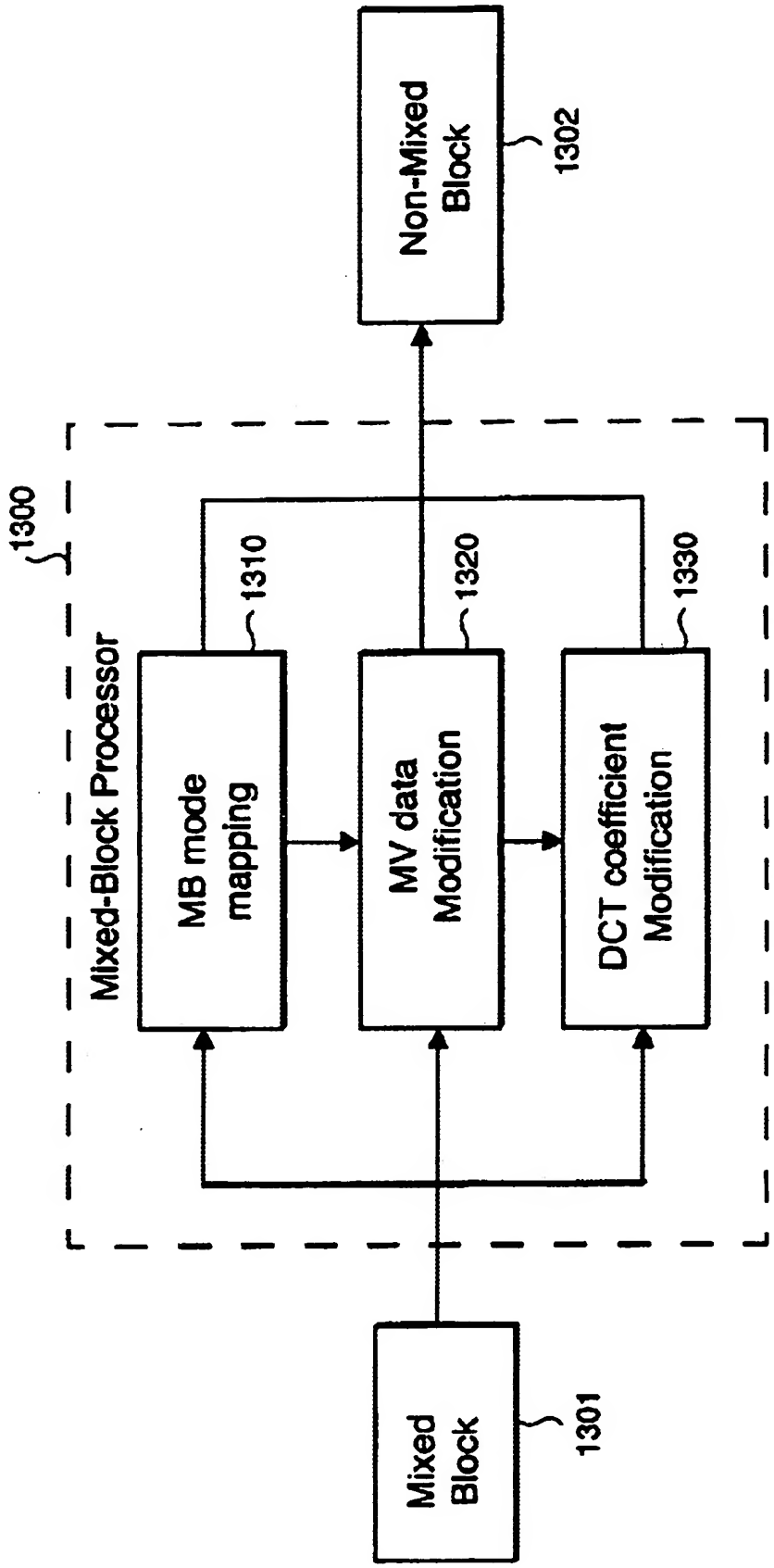

Non-Mixed Block
1302
1300
Mixed-Block Processor
1310
1320
1330
MB mode mapping
MV data Modification
DCT coefficient Modification
Mixed Block
1301

FIG. 13

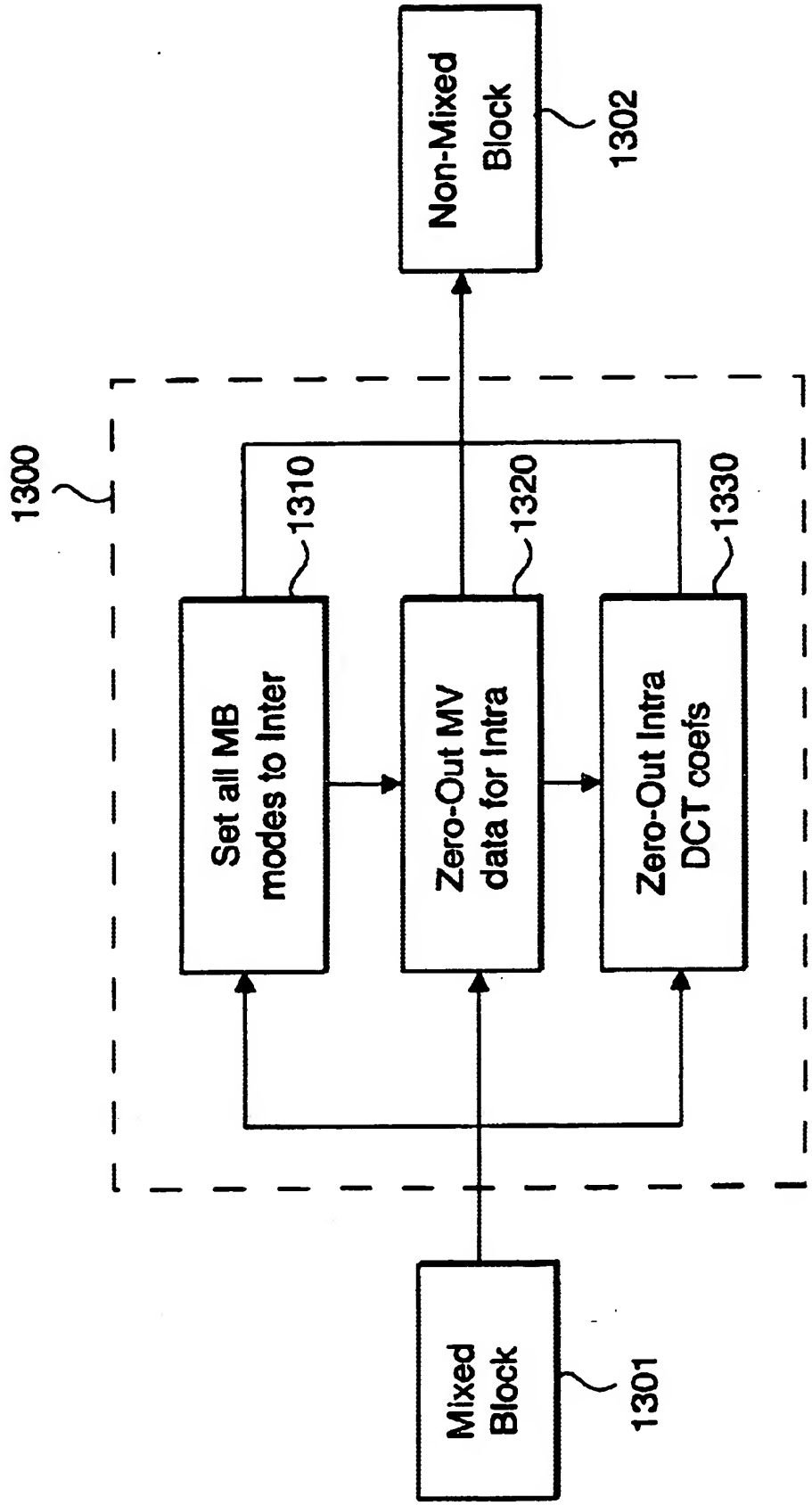

Non-Mixed Block
1302
1300
1310
1320
1330
Set all MB modes to Inter
Zero-Out MV data for Intra
Zero-Out Intra DCT coefs
Mixed Block
1301

FIG. 14A

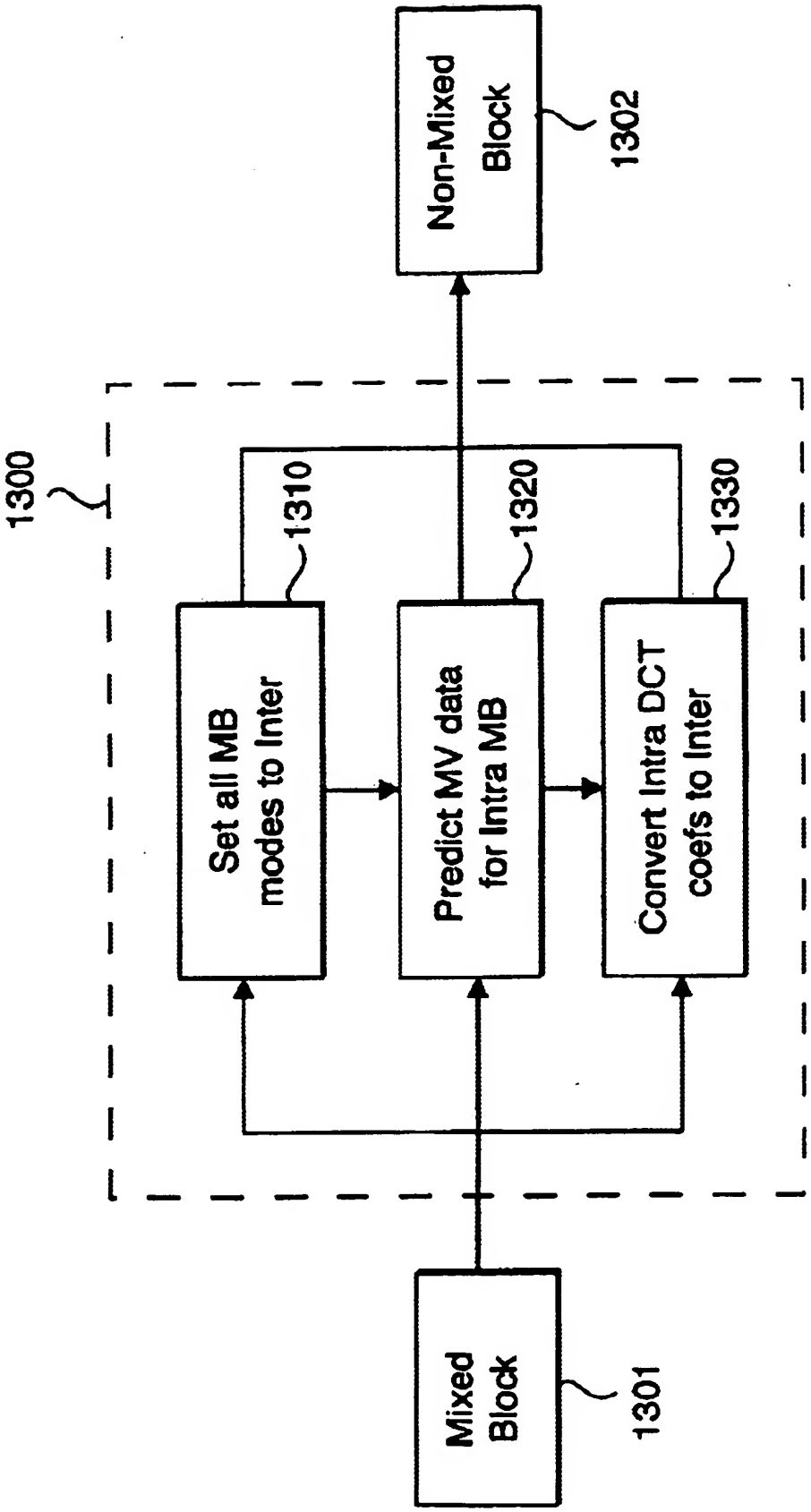

Non-Mixed Block
1302
1300
1310
1320
1330
Set all MB modes to Inter
Predict MV data for Intra MB
Convert Intra DCT coefs to Inter
Mixed Block
1301

FIG. 14B

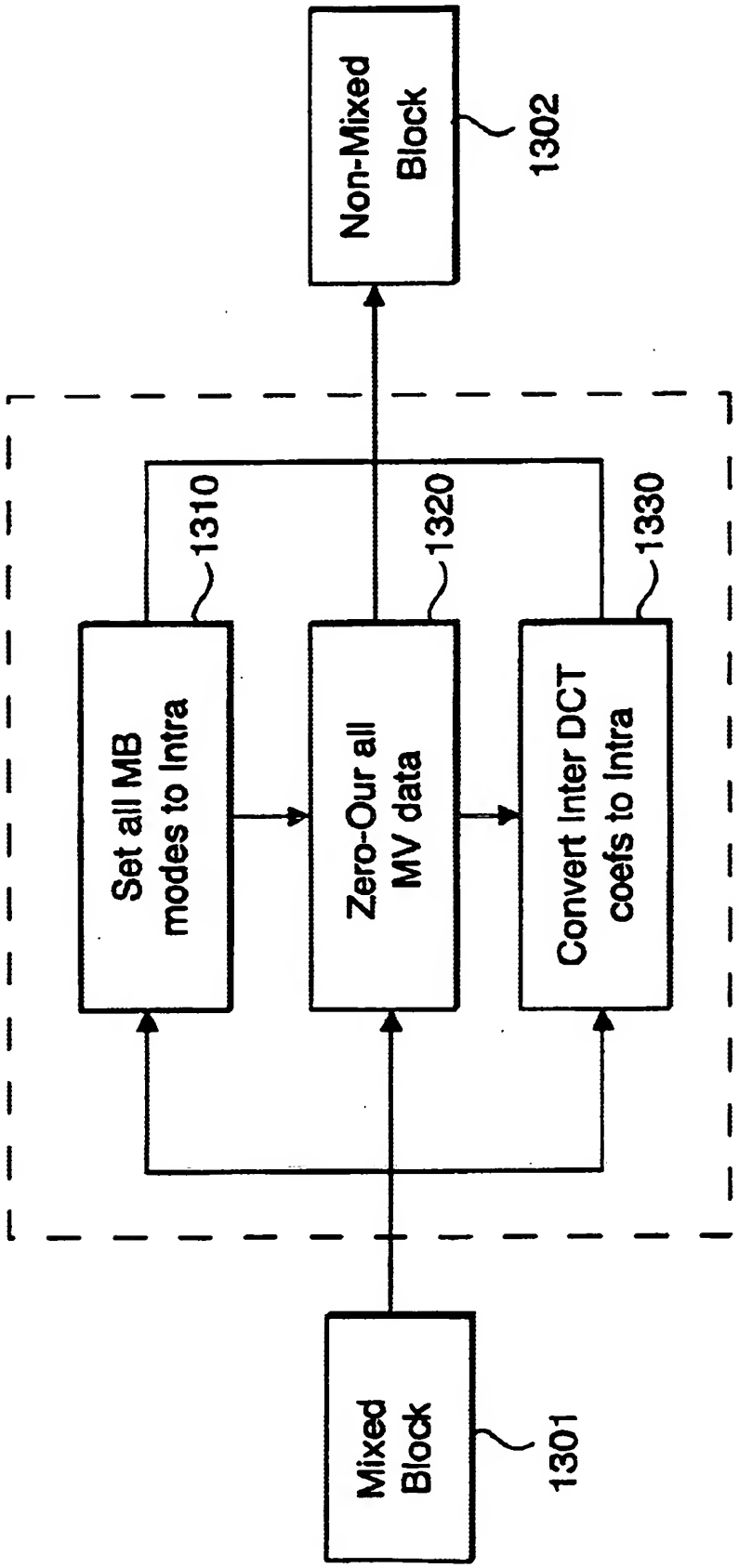

Non-Mixed Block
1302
1310
1320
1330
Set all MB modes to Intra
Zero-Our all MV data
Convert Inter DCT coefs to Intra
Mixed Block
1301

FIG. 14C

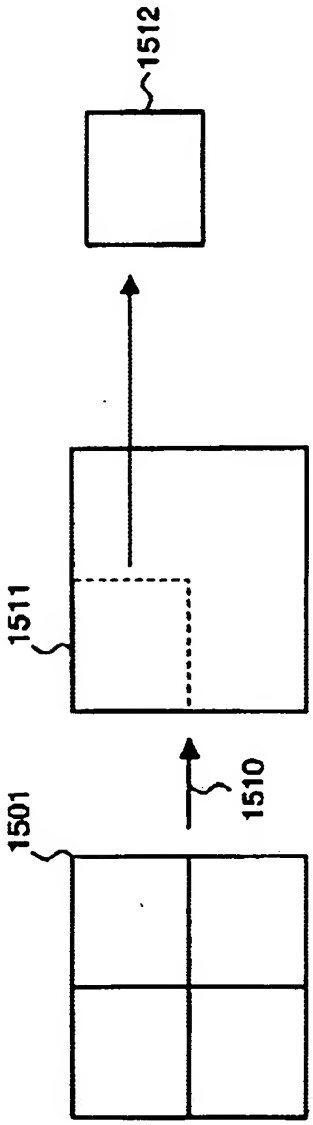


FIG. 15A
PRIOR ART

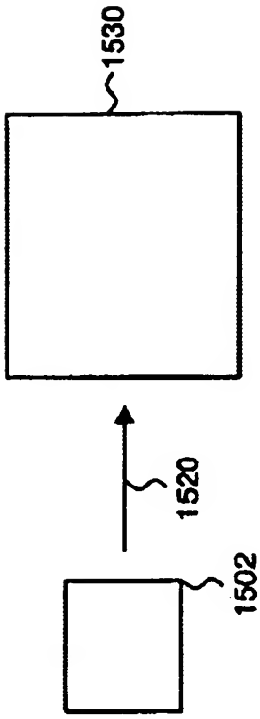


FIG. 15B

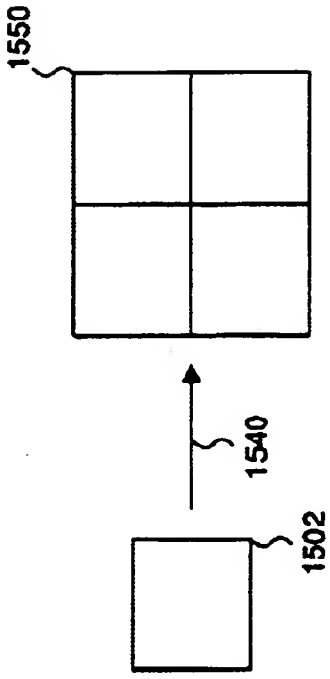


FIG. 15C

FIG. 15

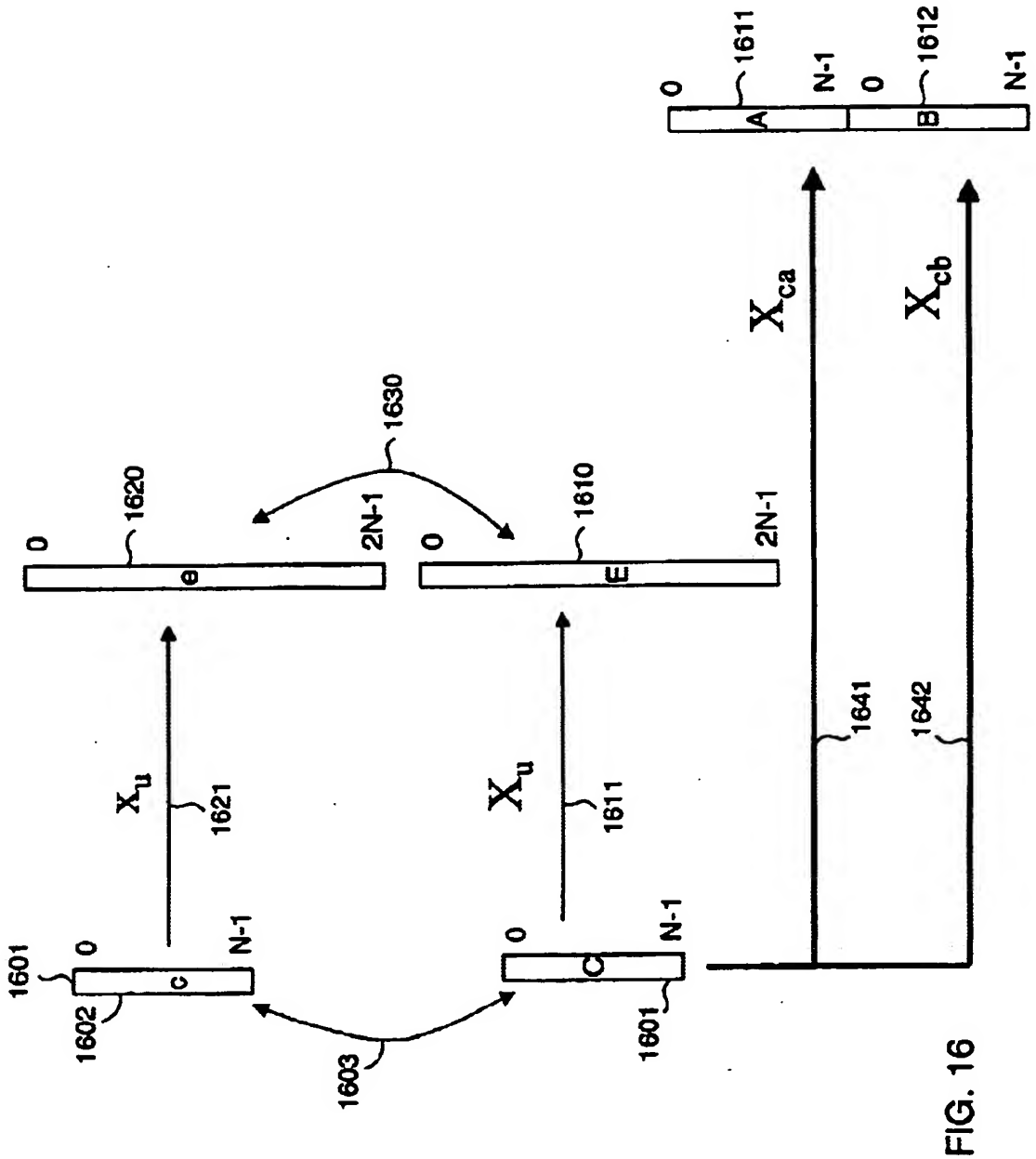

0
1611
N-1
A
0
1612
N-1
B
X_{ca}
X_{cb}
1630
0
1620
2N-1
e
0
1610
2N-1
E
X_u
1621
X_u
1611
1641
1642
1601
0
N-1
C
1602
1603
0
N-1
C
1601

FIG. 16

VIDEO TRANSCODER WITH SPATIAL RESOLUTION REDUCTION

FIELD OF THE INVENTION

This invention relates generally to the field of transcoding bitstreams, and more particularly to reducing spatial resolution while transcoding video bitstreams.

BACKGROUND OF THE INVENTION

Video compression enables the storing, transmitting, and processing of visual information with fewer storage, network, and processor resources. The most widely used video compression standards include MPEG-1 for storage and retrieval of moving pictures, MPEG-2 for digital television, and H.263 for video conferencing, see ISO/IEC 11172-2:1993, "Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbit/s—Part 2: Video," D. LeGall, "MPEG: A Video Compression Standard for Multimedia Applications," Communications of the ACM, Vol. 34, No. 4, pp. 46–58, 1991, ISO/IEC 13818-2:1996, "Information Technology—Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video," 1994, ITU-T SG XV, DRAFT H.263, "Video Coding for Low Bitrate Communication," 1996, ITU-T SG XVI, DRAFT13 H.263+ Q15-A-60 rev.0, "Video Coding for Low Bitrate Communication," 1997.

These standards are relatively low-level specifications that primarily deal with a spatial compression of images or frames, and the spatial and temporal compression of sequences of frames. As a common feature, these standards perform compression on a per frame basis. With these standards, one can achieve high compression ratios for a wide range of applications.

Newer video coding standards, such as MPEG-4 for multimedia applications, see ISO/IEC 14496-2:1999, "Information technology—coding of audio/visual objects, Part 2: Visual," allow arbitrary-shaped objects to be encoded and decoded as separate video object planes (VOP). The objects can be visual, audio, natural, synthetic, primitive, compound, or combinations thereof. Also, there is a significant amount of error resilience features built into this standard to allow for robust transmission across error-prone channels, such as wireless channels.

The emerging MPEG-4 standard is intended to enable multimedia applications, such as interactive video, where natural and synthetic materials are integrated, and where access is universal. In the context of video transmission, these compression standards are needed to reduce the amount of bandwidth on networks. The networks can be wireless or the Internet. In any case, the network has limited capacity, and contention for scarce resources should be minimized.

A great deal of effort has been placed on systems and methods that enable devices to transmit the content robustly and to adapt the quality of the content to the available network resources. When the content is encoded, it is sometimes necessary to further decode the bitstream before it can be transmitted through the network at a lower bit-rate or resolution.

As shown in FIG. **1**, this can be accomplished by a transcoder **100**. In a simplest implementation, the transcoder **100** includes a cascaded decoder **110** and encoder **120**. A compressed input bitstream **101** is fully decoded at an input bit-rate R_{in}, then encoded at an output bit-rate R_{out} **102** to produce the output bitstream **103**. Usually, the output rate is lower than the input rate. In practice, full decoding and full encoding in a transcoder is not done due to the high complexity of encoding the decoded bitstream.

Earlier work on MPEG-2 transcoding has been published by Sun et al., in "Architectures for MPEG compressed bitstream scaling," IEEE Transactions on Circuits and Systems for Video Technology, April 1996. There, four methods of rate reduction, with varying complexity and architecture, were described.

FIG. **2** shows a first example method **200**, which is referred to as an open-loop architecture. In this architecture, the input bitstream **201** is only partially decoded. More specifically, macroblocks of the input bitstream are variable-length decoded (VLD) **210** and inverse quantized **220** with a fine quantizer Q_1, to yield discrete cosine transform (DCT) coefficients. Given the desired output bit-rate **202**, the DCT blocks are a re-quantized by a coarser level quantizer Q_2 of the quantizer **230**. These re-quantized blocks are then variable-length coded (VLC) **240**, and a new output bitstream **203** at a lower rate is formed. This scheme is much simpler than the scheme shown in FIG. **1** because the motion vectors are re-used and an inverse DCT operation is not needed. Note, here the choice of Q_1 and Q_2 strictly depend on rate characteristics of the bitstream. Other factors, such as possibly, spatial characteristics of the bitstream are not considered.

FIG. **3** shows a second example method **300**. This method is referred to as a closed-loop architecture. In this method, the input video bitstream is again partially decoded, i.e., macroblocks of the input bitstream are variable-length decoded (VLD) **310**, and inverse quantized **320** with Q_1 to yield discrete cosine transform (DCT) coefficients **321**. In contrast to the first example method described above, correction DCT coefficients **332** are added **330** to the incoming DCT coefficients **321** to compensate for the mismatch produced by re-quantization. This correction improves the quality of the reference frames that will eventually be used for decoding. After the correction has been added, the newly formed blocks are re-quantized **340** with Q_2 to satisfy a new rate, and variable-length coded **350**, as before. Note, again Q_1 and Q_2 are rate based.

To obtain the correction component **332**, the re-quantized DCT coefficients are inverse quantized **360** and subtracted **370** from the original partially decoded DCT coefficients. This difference is transformed to the spatial domain via an I inverse DCT (IDCT) **365** and stored into a frame memory **380**. The motion vectors **381** associated with each incoming block are then used to recall the corresponding difference blocks, such as in motion compensation **290**. The corresponding blocks are then transformed via the DCT **332** to yield the correction component. A derivation of the method shown in FIG. **3** is described in "A frequency domain video transcoder for dynamic bit-rate reduction of MPEG-2 bitstreams," by Assuncao et al., IEEE Transactions on Circuits and Systems for Video Technology, pp. 953–957, 1998.

Assuncao et al. also described an alternate method for the same task. In the alternative method, they used a motion compensation (MC) loop operating in the frequency domain for drift compensation. Approximate matrices were derived for fast computation of the MC blocks in the frequency domain. A Lagrangian optimization was used to calculate the best quantizer scales for transcoding. That alternative method removed the need for the IDCT/DCT components.

According to prior art compression standards, the number of bits allocated for encoding texture information is con-

trolled by a quantization parameter (QP). The above methods are similar in that changing the QP based on information that is contained in the original bitstream reduces the rate of texture bits. For an efficient implementation, the information is usually extracted directly from the compressed domain and can include measures that relate to the motion of macroblocks or residual energy of DCT blocks. The methods describes above are only applicable for bit-rate reduction.

Besides bit-rate reduction, other types of transformation of the bitstream can also be performed. For example, object-based transformations have been described in U.S. patent application Ser. No. 09/504,323, "Object-Based Bitstream Transcoder," filed on Feb. 14, 2000 by Vetro et al. Transformations on the spatial resolution have been described in "Heterogeneous video transcoding to lower spatio-temporal resolutions, and different encoding formats," IEEE Transaction on Multimedia, June 2000, by Shanableh and Ghanbari.

It should be noted these methods produce bitstreams at a reduced spatial resolution reduction that lack quality, or are accomplished with high complexity. Also, proper consideration has not been given to the means by which reconstructed macroblocks are formed. This can impact both the quality and complexity, and is especially important when considering reduction factors different than two. Moreover, these methods do not specify any architectural details. Most of the attention is spent on various means of scaling motion vectors by a factor of two.

FIG. 4 shows the details of a method **400** for transcoding an input bitstream to an output bitstream **402** at a lower spatial resolution. This method is an extension of the method shown in FIG. **1**, but with the details of the decoder **110** and encoder **120** shown, and a down-sampling block **410** between the decoding and encoding processes. The decoder **110** performs a partial decoding of the bitstream. The down-sampler reduces the spatial resolution of groups of partially macroblocks. Motion compensation **420** in the decoder uses the full-resolution motion vectors mv_f **421**, while motion compensation **430** in the encoder uses low-resolution motion vectors mv_r **431**. The low-resolution motion vectors are either estimated from the down-sampled spatial domain frames y_n^1 **403**, or mapped from the full-resolution motion vectors. Further detail of the transcoder **400** are described below.

FIG. **5** shows the details of an open-loop method **500** for transcoding an input bitstream **501** to an output bitstream **502** at a lower spatial resolution. In this method, the video bitstream is again partially decoded, i.e., macroblocks of the input bitstream are variable-length decoded (VLD) **510** and inverse quantized **520** to yield discrete cosine transform (DCT) coefficients, these steps are well known.

The DCT macroblocks are then down-sampled **530** by a factor of two by masking the high frequency coefficients of each 8×8 (2^3×2^3)luminance block in the 16×16 (2^4×2^4) macroblock to yield four 4×4 DCT blocks, see U.S. Pat. No. 5,262,854, "Low-resolution HDTV receivers," issued to Ng on Nov. 16, 1993. In other words, down-sampling turns a group of blocks, for example four, into a group of four blocks of a smaller size.

By performing down-sampling in the transcoder, the transcoder must take additional steps to re-form a compliant 16×16 macroblock, which involves transformation back to the spatial domain, then again to the DCT domain. After the down-sampling, blocks are re-quantized **540** using the same quantization level, and then variable length coded **550**. No methods have been described to perform rate control on the reduced resolution blocks.

To perform motion vector mapping **560** from full **559** to reduced **561** motion vectors, several methods suitable for frame-based motion vectors have been described in the prior art. To map from four frame-based motion vectors, i.e., one for each macroblock in a group, to one motion vector for the newly formed 16×16 macroblock, simple averaging or median filters can be applied. This is referred to as a 4:1 mapping.

However, certain compression standards, such as MPEG-4 and H.263, support advanced prediction modes that allow one motion vector per 8×8 block. In this case, each motion vector is mapped from a 16×16 macroblock in the original resolution to an 8×8 block in the reduced resolution macroblock. This is referred to as a 1:1 mapping.

FIG. **6** shows possible mappings **600** of motion vector from a group of four 16×16 macroblocks **601** to either one 16×16 macroblock **602** or four 8×8 macroblocks **603**. It is inefficient to always use the 1:1 mapping because more bits are used to code four motion vectors. Also, in general, the extension to field-based motion vectors for interlaced images is non-trivial. Given the down-sampled DCT coefficients and mapped motion vectors, the data are subject to variable length coding and the reduced resolution bitstream can be formed as is well known.

It is desired to provide a method for transcoding bitstreams that overcomes the problems of the prior art methods for spatial resolution reduction. Furthermore, it is desired to provide a balance between complexity and quality in the transcoder. Furthermore it is desired to compensate for drift, and provide better up-sampling techniques during the transcoding.

SUMMARY OF THE INVENTION

A method up-samples a compressed bitstream. The compressed bitstream is partially decoding to produce macroblocks. Each macroblock has DCT coefficients according to a predetermined dimensionality of the macroblock.

DCT filters are applied to the DCT coefficients of each macroblock to generate up-sampled macroblocks for each macroblock, there is one up-sampled macroblock generated by each filter. Each generated up-sampled macroblock has the predetermined dimensionality.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram of a prior art cascaded transcoder;

FIG. **2** is a block diagram of a prior art open-loop transcoder for bit-rate reduction;

FIG. **3** is a block diagram of a prior art closed-loop transcoder for bit-rate reduction;

FIG. **4** is a block diagram of a prior art cascaded transcoder for spatial resolution reduction;

FIG. **5** is a block diagram of a prior art open-loop transcoder for spatial resolution reduction;

FIG. **6** is a block diagram of prior art motion vector mapping;

FIG. **7** is a block diagram of a first application transcoding a bitstream to a reduced spatial resolution according to the invention;

FIG. **8** is a block diagram of a second application transcoding a bitstream to a reduced spatial resolution according to the invention;

FIG. 9 is a block diagram of an open-loop transcoder for spatial resolution reduction according to the invention;

FIG. 10 is a block diagram of a first closed-loop transcoder for spatial resolution reduction with drift compensation in the reduced resolution according to the invention;

FIG. 11*a* is a block diagram of a second closed-loop transcoder for spatial resolution reduction with drift compensation in the original resolution according to the invention;

FIG. 11*b* is a block diagram of a third closed-loop transcoder for spatial resolution reduction with drift compensation in the original resolution according to the invention;

FIG. 12 is an example of a group of macroblocks containing macroblock modes, DCT coefficient data, and corresponding motion vector data;

FIG. 13 is a block diagram of a group of blocks processor according to the invention;

FIG. 14A is a block diagram of a first method for group of blocks processing according to the invention;

FIG. 14B is block diagram of a second method for group of blocks processing according to the invention;

FIG. 14C is a block diagram of a third method for a group of blocks processing according to the invention;

FIG. 15A illustrates a prior art concept of down-sampling in the DCT or spatial domain;

FIG. 15B is a block diagram of prior art up-sampling in the DCT or spatial domain;

FIG. 15C is a block diagram of up-sampling in the DCT domain according to the invention; and

FIG. 16 is a diagram of up-sampling in the DCT domain according to the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Introduction

The invention provides a system and method for transcoding compressed bitstreams of digital video signals to a reduced spatial resolution with minimum drift. First, several applications for content distribution that can use the transcoder according to the invention are described. Next, an analysis of a basic method for generating a bitstream at a lower spatial resolution is provided. Based on this analysis, several alternatives to the base method and the corresponding architectures that are associated with each alternative are described.

A first alternative, see FIG. 9, uses an open-loop architecture, while the other three alternatives, FIGS. 10 and 11*a–b*, correspond to closed-loop architectures that provide a means of compensating drift incurred by down-sampling, re-quantization and motion vector truncation. One of the closed-loop architectures performs this compensation in the reduced resolution, while the others perform this compensation in the original resolution in the DCT domain for better quality.

As will be described in greater detail below, the open-loop architecture of FIG. 9 is of low complexity. There is no reconstruction loop, no DCT/IDCT blocks, no frame store, and the quality is reasonable for low picture resolution, and bit-rates. This architecture is suitable for Internet applications and software implementations. The first closed-loop architecture of FIG. 10 is also of moderate complexity. It includes a reconstruction loop, IDCT/DCT blocks, and a frame store. Here, the quality can be improved with drift compensation in reduced resolution domain. The second closed-loop architecture of FIG. 11*a* is of moderate complexity. It includes a reconstruction loop, IDCT/DCT blocks, and a frame store. The quality can be improved with drift compensation in the original resolution domain, and does require up-sampling of the reduced resolution frames. The third closed loop architecture uses a correction signal obtained in the reduced resolution domain.

To support the architectures according to the present invention, several additional techniques for processing blocks that would otherwise have groups of macroblock with "mixed" modes at the reduced resolution are also described.

A group of blocks, e.g., four, to be down-sampled is considered a "mixed" block when the group of blocks to be down-sampled contains blocks coded in both intra- and inter-modes. In the MPEG standards I-frames contain only macroblocks coded according to the intra-mode, and P-frames can include intra- and inter-mode coded blocks. These modes need to be respected, particularly while down-sampling, otherwise the quality of the output can be degraded.

Also, methods for drift-compensation and up-sampling DCT based data are described. These methods are useful for the second and third closed-loop architectures so that operations after the up-sampling can be performed properly and without additional conversion steps.

Applications for Reduced Spatial Resolution Transcoding

The primary target application for the present invention is the distribution of digital television (DTV) broadcast and Internet content to devices with low-resolution displays, such as wireless telephones, pagers, and personal digital assistance. MPEG-2 is currently used as the compression format for DTV broadcast and DVD recording, and MPEG-1 content is available over the Internet.

Because MPEG-4 has been adopted as the compression format for video transmission over mobile networks, the present invention deals with methods for transcoding MPEG-1/2 content to lower resolution MPEG-4 content.

FIG. 7 shows a first example of a multimedia content distribution system **700** that uses the invention. The system **700** includes an adaptive server **701** connected to clients **702** via an external network **703**. As a characteristics the clients have small-sized displays or are connected by low bit-rate channels. Therefore, there is a need to reduce the resolution of any content distributed to the clients **702**.

Input source multimedia content **704** is stored in a database **710**. The content is subject to a feature extraction and an indexing process **720**. A database server **740** allows the clients **702** to browse the content of the database **710** and to make requests for specific content. A search engine **730** can be used to locate multimedia content. After the desired content has been located, the database server **740** forwards the multimedia content to a transcoder **750** according to the invention.

The transcoder **750** reads network and client characteristics. If the spatial resolution of the content is higher than the display characteristics of the client, then the method according to the invention is used to reduce the resolution of the content to match the display characteristics of the client. Also, if the bit-rate on the network channel is less than the bit-rate of the content, the invention can also be used.

FIG. 8 shows a second example of a content distribution system **800**. The system **800** includes a local "home" network **801**, the external network **703**, a broadcast network **803**, and the adaptive server **701** as described for FIG. 7. In this application, high-quality input source content **804** can

be transported to clients **805** connected to the home network **801** via the broadcast network **803**, e.g., cable, terrestrial or satellite. The content is received by a set-top box or gateway **820** and stored into a local memory or hard-disk drive (HDD) **830**. The received content can be distributed to the clients **805** within the home. In addition, the content can be transcoded **850** to accommodate any clients that do not have the capability to decode/display the full resolution content. This can be the case when a high-definition television (HDTV) bitstream is received for a standard-definition television set. Therefore, the content should be transcoded to satisfy client capabilities within the home.

Moreover, if access to the content stored on the HDD **830** is desired by a low-resolution external client **806** via the external network **802**, then the transcoder **850** can also be used to deliver low-resolution multimedia content to this client.

Analysis of Base Method

In order to design a transcoder with varying complexity and quality, the signals generated by the method of FIG. **4** are further described and analyzed. With regard to notation in the equations, lowercase variables indicate spatial domain signals, while uppercase variables represent the equivalent signal in the DCT domain. The subscripts on the variables indicates time, while a superscript equal to one denotes a signal that has drift and a superscript equal to two denotes a signal that is drift free. The drift is introduced through lossy processes, such as re-quantization, motion vector truncation or down-sampling. A method for drift compensation is described below.

I-frames

Because there is no motion compensated prediction for I-frames, i.e.,

$$x_n^1=e_n^1, \tag{1}$$

the signal is down-sampled **410**,

$$y_n^1=D(x_n^1). \tag{2}$$

Then, in the encoder **120**,

$$g_n^2=y_n^1. \tag{3}$$

The signal g_n^2 is subject to the DCT **440**, then quantized **450** with quantization parameter Q_2. The quantized signal c_{out} is variable length coded **460** and written to the transcoded bitstream **402**. As part of the motion compensation loop in the encoder, c_{out} is inverse quantized **470** and subject to the IDCT **480**. The reduced resolution reference signal y_n^2 **481** is stored into the frame buffer **490** as the reference signal for future frame predictions.

P-frames

In the case of P-frames, the identity

$$x_n^1=e_n^1+M_f(x_{n-1}^1) \tag{4}$$

yields the reconstructed full-resolution picture. As with the I-frame, this signal is then down-converted via equation (2). Then, the reduced-resolution residual is generated according to

$$g_n^2=y_n^1-M_r(y_{n-1}^2), \tag{5}$$

which is equivalently expressed as,

$$g_n^2=D(e_n^1)+D(x_{n-1}^1))-M_r(y_{n-1}^2). \tag{6}$$

The signal given by equation (6) represents the reference signal that the architectures described by this invention approximate. It should be emphasized that the complexity in generating this reference signal is high and is desired to approximate the quality, while achieving significant complexity reduction.

Open-Loop Architecture

Give the approximations,

$$y_{n-1}^2=y_{n-1}^1 \tag{7a}$$

$$D(M_f(x_{n-1}^1))=M_r(D(x_{n-1}^1))=M_r(y_{n-1}^1) \tag{7b}$$

the reduced resolution residual signal in equation (6) is expressed as,

$$g_n^2=D(e_n^1). \tag{8}$$

The above equation suggests the open-loop architecture for a transcoder **900** as shown in FIG. **9**.

In the transcoder **900**, the incoming bitstream **901** signal is variable length decoded **910** to generate inverse quantized DCT coefficients **911**, and full resolution motion vectors, mv_f **902**. The full-resolution motion vectors are mapped by the MV mapping **920** to reduced-resolution motion vectors, mv_r **903**. The quantized DCT coefficients **911** are inverse quantized, with quantizer Q_1 **930**, to yield signal E_n^1 **931**. This signal is then subject to a group of blocks processor **1300** as described in greater detail below. The output of the processor **1300** is down-sampled **950** to produce signal G^2_n **951**. After down-sampling, the signal is quantized with quantizer Q_2 **960**. Finally, the reduced resolution re-quantized DCT coefficients and motion vectors are variable length coded **970** and written to the transcoded output bitstream **902**.

The details and preferred embodiments of the group of blocks processor **1300** are described below, but briefly, the purpose of the group of blocks processor is to pre-process selected groups of macroblocks to ensure that the down-sampling process **950** will not generate groups of macroblocks in which its sub-blocks have different coding modes, e.g., both inter-and intra-blocks. Mixed coding modes within a macroblock are not supported by any known video coding standards.

Drift Compensation in Reduced Resolution

Given only the approximation given by equation (7b), the reduced resolution residual signal in equation (6) is expressed as,

$$g_n^2=D(e_n^1)+M_r(y_{n-1}^1-y_{n-1}^2) \tag{9}$$

The above equation suggests the closed-loop architecture **1000** shown in FIG. **10**, which compensates for drift in the reduced resolution.

In this architecture, the incoming signal **1001** is variable length decoded **1010** to yield quantized DCT coefficients **1011** and full resolution motion vectors mv_f **1012**. The full-resolution motion vectors **1012** are mapped by the MV mapping **1020** to yield a set of reduced-resolution motion vectors, mv_r **1021**. The quantized DCT coefficients are inverse quantized **1030**, with quantizer Q_1 to yield signal E_n^1 **1031**. This signal is then subject to the group of blocks processor **1300** and down-sampled **1050**. After down-sampling **1050**, a reduced-resolution drift-compensating signal **1051** is added **1060** to the low-resolution residual **1052** in the DCT domain.

The signal **1061** is quantized with spatial quantizer Q_2 **1070**. Finally, the reduced resolution re-quantized DCT coefficients **1071** and motion vectors **1021** are variable length coded **1080** to generate the output transcoded bitstream **1002**.

The reference frame from which the reduced-resolution drift-compensating signal is generated is obtained by an inverse quantization **1090** of the re-quantizer residual G^2 **1071**, which is then subtracted **1092** from the down-sampled residual G_n^2 **1052**. This difference signal is subject to the IDCT **1094** and added **1095** to the low-resolution predictive component **1096** of the previous macroblock stored in the frame store **1091**. This new signal represents the difference $(y_{n-1}^1 y_{n-1}^2)$ **1097** and is used as the reference for low-resolution motion compensation for the current block.

Given the stored reference signal, low-resolution motion compensation **1098** is performed and the prediction is subject to the DCT **1099**. This DCT-domain signal is the reduced-resolution drift-compensating signal **1051**. This operation is performed on a macroblock-by-macroblock basis using the set of low-resolution motion vectors, mv_r **1021**.

First Method of Drift Compensation in Original Resolution

For an approximation,

$$M_f(y_{n-1}^2)=D(M_f(U(y_{n-1}^2)))=D(M_f(x_{n-1}^2)), \quad (10)$$

the reduced resolution residual signal in equation (6) is expressed as,

$$g_n^2=D(e_n^1)+M_f(x_{n-1}^1-x_{n-1}^2). \quad (11)$$

The above equation suggests the closed-loop architecture **1100** shown in FIG. **11**, which compensates for drift in the original resolution bitstream.

In this architecture, the incoming signal **1001** is variable length decoded **1110** to yield quantized DCT coefficients **1111**, and full resolution motion vectors, mv_f **1112**. The quantized DCT coefficients **1111** are inverse quantized **1130**, with quantizer Q_1, to yield signal E_n^1 **1131**. This signal is then subject to the group of blocks processor **1300**. After group of blocks processing **1300**, an original-resolution drift-compensating signal **1151** is added **1160** to the residual **1141** in the DCT domain. The signal **1162** is then down-sampled **1150**, and quantized **1170** with quantizer Q_2. Finally, the reduced resolution re-quantized DCT coefficients **1171**, and motion vectors **1121** are variable length coded **1180**, and written to the transcoded bitstream **1102**.

The reference frame from which the original-resolution drift-compensating signal **1151** is generated by an inverse quantization **1190** of the re-quantizer residual G_n^2 **1171**, which is then up-sampled **1191**. Here, after the up-sampling the up-sampled signal is subtracted **1192** from the original resolution residual **1161**. This difference signal is subject to the IDCT **1194**, and added **1195** to the original-resolution predictive component **1196** of the previous macroblock. This new signal represents the difference $(x_{n-1}^1-x_{n-1}^2)$ **1197**, and is used as the reference for motion compensation of the current macroblock in the original resolution.

Given the reference signal stored in the frame buffer **1181**, original-resolution motion compensation **1198** is performed, and the prediction is subject to the DCT **1199**. This DCT-domain signal is the original-resolution drift-compensating signal **1151**. This operation is performed on a macroblock-by-macroblock basis using the set of original-resolution motion vectors, mv_f **1121**.

Second Method of Drift Compensation in Original Resolution

FIG. **11***b* shows an alternative embodiment of the closed loop architecture of FIG. **11***a*. Here, the output of the inverse quantization **1190** of the re-quantizer residual G_n^2 **1172** is subtracted **1192** from the reduced resolution signal before up-sampling **1191**.

Both drift compensating architectures in the original resolution do not use the motion vector approximations in generating the drift compensating signal **1151**. This is accomplished by the use of up-sampling **1191**. The two alternative architectures mainly differ in the choice of signals that are used to generate the difference signal. In the first method, the difference signal represents error due to re-quantization and resolution conversion, while the difference signal in the second method only considers the error due to re-quantization.

Because the up-sampled signal is not considered in the future decoding of the transcoded bitstream, it is reasonable to exclude any error measured by consecutive down-sampling and up-sampling in the drift compensation signal. However, up-sampling is still employed for two reasons: to make use of the full-resolution motion vectors **1121** to avoid any further approximation, and so that the drift compensating signal is in the original resolution and can be added **1160** to the incoming residual **1161** before down-sampling **1150**.

Mixed Block Processor

The purpose of the group of blocks processor **1300** is to pre-process selected macroblocks to ensure that the down-sampling process do not generate macroblocks in which its sub-blocks have different coding modes, e.g., inter- and intra-blocks. Mixed coding modes within macroblocks are not supported by any known video coding standards.

FIG. **12** shows an example of a group of macroblocks **1201** that can lead to a group of blocks **1202** in the reduced resolution after transcoding **1203**. Here, there are three inter-mode blocks, and one intra-mode block. Note, the motion vector (MV) for the intra-mode block is zero. Determining whether a particular group of blocks is a mixed group, or not, depends only on the macroblock mode. The group of blocks processor **1300** considers groups of four macroblocks **1201** that form a single macroblock **1202** in the reduced resolution. In other words, for the luminance component, MB(0) **1210** corresponds to sub-block b(0) **1220** in the reduced resolution macroblock **1202**, and similarly, MB(1) **1211** will correspond to b(1) **1221**, MB(k) **1212** corresponds to b(2) **1222**, and MB(k+1) **1213** corresponds to b(3) **1223**, where k is the number of macroblocks per row in the original resolution. Chrominance components are handled in a similar manner that is consistent with luminance modes.

A group of MB modes determine whether the group of blocks processor **1300** should process a particular MB. The group of blocks is processed if the group contains at least one intra-mode block, and at least one inter-mode block. After a macroblock is selected, its DCT coefficients and motion vector data are subject to modification.

FIG. **1300** shows the components of the group of blocks processor **1300**. For a selected group of mixed blocks **1301**, the group of blocks processor performs mode mapping **1310**, motion vector modification **1320**, and DCT coefficient modification **1330** to produce an output non-mixed block **1302**. Given that the group of blocks **1301** has been identified, the modes of the macroblocks are modified so that all macroblocks are identical. This is done according to a pre-specified strategy to match the modes of each sub-block in a reduced resolution block.

In accordance with the chosen mode mapping, the MV data are then subject to modification **1320**. Possible modifications that agree with corresponding mode mappings are described in detail below for FIGS. **14**A–C. Finally, given both the new MB mode and the MV data, the corresponding DCT coefficients are also modified **1330** to agree with the mapping.

In a first embodiment of the group of blocks processor as shown in FIG. **14A**, the MB modes of the group of blocks **1301** are modified to be inter-mode by the mode mapping **1310**. Therefore, the MV data for the intra-blocks are reset to zero by the motion vector processing, and the DCT coefficients corresponding to intra-blocks are also reset to zero by the DCT processing **1330**. In this way, such sub-blocks that have been converted are replicated with data from the corresponding block in the reference frame.

In a second embodiment of the group of blocks processor as shown in FIG. **14B**, the MB modes of the groups of mixed block are modified to be to inter-mode by the mapping **1310**. However, in contrast to the first preferred embodiment, the MV data for intra-MB's are predicted. The prediction is based on the data in neighboring blocks, which can include both texture and motion data. Based on this predicted motion vector, a new residual for the modified block is calculated. The final step **1320** resets the inter-DCT coefficients to intra-DCT coefficients.

In a third embodiment shown in FIG. **14C**, the MB modes of the grouped of blocks are modified **1310** to intra-mode. In this case, there is no motion information associated with the reduced-resolution macroblock, therefore all associated motion vector data are reset **1320** to zero. This is necessary to perform in the transcoder because the motion vectors of neighboring blocks are predicted from the motion of this block. To ensure proper reconstruction in the decoder, the MV data for the group of blocks must be reset to zero in the transcoder. The final step **1330** generates intra-DCT coefficients to replace the inter-DCT coefficients, as above.

It should be noted that to implement the second and third embodiments described above, a decoding loop that reconstructs to full-resolution can be used. This reconstructed data can be used as a reference to convert the DCT coefficients between intra- and inter-modes, or inter- and intra-modes. However, the use of such a decoding loop is not required. Other implementations can perform the conversions within the drift compensating loops.

For a sequence of frames with a small amount of motion, and a low-level of detail the low complexity strategy of FIG. **14A** can be used. Otherwise, the equally complex strategies of either FIG. **14***b* or FIG. **14***c* should be used. The strategy of FIG. **14***c* provides the best quality.

Drift Compensation with Block Processing

It should be noted that the group of block processor **1300** can also be used to control or minimize drift. Because intra coded blocks are not subject to drift, the conversion of inter-coded blocks to intra-coded blocks lessens the impact of drift.

As a first step **1350** of FIG. **14C**, the amount of drift in the compressed bitstream is measured. In the closed-loop architectures, the drift can be measured according to the energy of the difference signal generated by **1092** and **1192** or the drift compensating signal stored in **1091** and **1191**. Computing the energy of a signal is a well-known method. The energy that is computed accounts for various approximations, including re-quantization, down-sampling and motion vector truncation.

Another method for computing the drift, which is also applicable to open-loop architectures, estimates the error incurred by truncated motion vectors. It is known that half-pixel motion vectors in the original resolution lead to large reconstruction errors when the resolution is reduced. Full-pixel motion vectors are not subject to such errors because they can still be mapped correctly to half-pixel locations. Given this, one possibility to measure the drift is to record the percentage of half-pixel motion vectors. However, because the impact of the motion vector approximation depends on the complexity of the content, another possibility is that the measured drift be a function of the residual components that are associated with blocks having half-pixel motion vectors.

The methods that use the energy of the difference signal and motion vector data to measure drift can be used in combination, and can also be considered over sub-regions in the frame. Considering sub-regions in the frame is advantageous because the location of macroblocks that benefit most by drift compensation method can be identified. To use these methods in combination, the drift is measured by the energy of the difference signal, or drift compensating signal for macroblocks having half-pixel motion vectors in the original resolution.

As a second step, the measured value of drift is translated into an "intra refresh rate" **1351** that is used as input to the group of blocks processor **1300**. Controlling the percentage of intra-coded blocks has been considered in the prior art for encoding of video for error-resilient transmission, see for example "Analysis of Video Transmission over Lossy Channels," Journal of Selected Areas of Communications, by Stuhlmuller, et al, 2000. In that work, a back-channel from the receiver to the encoder is assumed to communicate the amount of loss incurred by the transmission channel, and the encoding of intra-coded blocks is performed directly from the source to prevent error propagation due to lost data in a predictive coding scheme.

In contrast, the invention generates new intra-blocks in the compressed domain for an already encoded video, and the conversion from inter- to intra-mode is accomplished by the group of blocks processor **1300**.

If the drift exceeds a threshold amount of drift, the group of blocks processor **1300** of FIG. **14***c* is invoked to convert an inter-mode block to an intra-mode block. In this case, the conversion is be performed at a fixed and pre-specified intra refresh rate. Alternatively, conversion can be done at an intra refresh rate that is proportional to the amount of drift measured. Also, rate-distortion characteristics of the signal can be taken into account to make appropriate trade-offs between the intra refresh rate and quantizers used for coding intra and inter blocks.

It should be noted that the invention generates new intra-blocks in the compressed domain, and this form of drift compensation can be performed in any transcoder with or without resolution reduction.

Down-Sampling

Any down-sampling method can be used by the transcoder according to the invention. However, the preferred down-sampling method is according to U.S. Pat. No. 5,855,151, "Method and apparatus for down-converting a digital signal," issued on Nov 10, 1998 to Sun et al, incorporated herein by reference.

The concept of this down-sampling method is shown in FIG. **15A**. A group includes four $2^N \times 2^N$ DCT blocks **1501**. That is, the size of the group is $2^{N+1} \times 2^{N+1}$. A "frequency synthesis" or filtering **1510** is applied to the group of blocks to generate a single $2^N \times 2^N$ DCT block **1511**. From this synthesized block, a down-sampled DCT block **1512** can be extracted.

This operation has been described for the DCT domain using 2D operations, but the operations can also be performed using separable 1D filters. Also, the operations can be completely performed in the spatial domain. Equivalent spatial domain filters can be derived using the methods described in U.S. patent application Ser. No. 09/035,969, "Three layer scalable decoder and method of decoding," filed on Mar. 6, 1998 by Vetro et al, incorporated herein by reference.

The main advantage of using the down-sampling method in the transcoder according to the invention is that correct dimension of sub-blocks in the macroblock are obtained directly, e.g., from four 8×8 DCT blocks, a single 8×8 block can be formed. On the other hand, alternate prior art methods for down-sampling produce down-sampled data in a dimension that does not equal the required dimension of the outgoing sub-block of a macroblock, e.g., from four 8×8 DCT blocks, a four 4×4 DCT blocks is obtained. Then, an additional step is needed to compose a single 8×8 DCT block.

The above filters are useful components to efficiently implement the architecture shown in FIG. **11** that requires up-sampling. More generally, the filters derived here can be applied to any system that requires arithmetic operations on up-sampled DCT data, with or without resolution reduction or drift compensation.

Up-Sampling

Any means of prior art up-sampling can be used in the present invention. However, Vetro, et al., in U.S. patent application "Three layer scalable decoder and method of decoding," see above, states that the optimal up-sampling method is dependent on the method of down-sampling. Therefore, the use an up-sampling filters x_u that corresponds to the down-sampling filters x_d is preferred, where the relation between the two filters is given by,

$$x_u = x_d^T (x_d x_d^T)^{-1} \tag{12}$$

There are two problems associated with the filters derived from the above equations. First, the filters are only applicable in the spatial domain filters because the DCT filters are not invertable. But, this is a minor problem because the corresponding spatial domain filters can be derived, then converted to the DCT-domain.

However, the second problem is that the up-sampling filters obtained in this way correspond to the process shown in FIG. **15B**. In this process, for example, an $2^N \times 2^N$ block **1502** is up-sampled **1520** to a single $2^{N+1} \times 2^{N+1}$ block **1530**. If up-sampling is performed entirely in the spatial domain, there is no problem. However, if the up-sampling is performed in the DCT domain, one has a $2^{N+1} \times 2^{N+1}$ DCT block to deal with, i.e., with one DC component. This is not suitable for operations that require the up-sampled DCT block to be in standard MB format, i.e., four $2^N \times 2^N$ DCT blocks, where N is 4. That is, the up-sampled blocks have the same format or dimensionality as the original blocks, there just are more of them.

The above method of up-sampling in the DCT domain is not suitable for use in the transcoder described in this invention. In FIG. **11***a*, up-sampled DCT data are subtracted from DCT data output from the mixed block processor **1300**. The two DCT data of the two blocks must have the same format. Therefore, a filter that can perform the up-sampling illustrated in FIG. **15C** is required. Here, the single $2^N \times 2^N$ block **1502** is up-sampled **1540** to four $2^N \times 2^N$ blocks **1550**. Because such a filter has not yet been considered and does not exist in the known prior art, an expression for the ID case is derived in the following.

With regard to notation in the following equations, lowercase variables indicate spatial domain signals, while uppercase variables represent the equivalent signal in the DCT domain.

As illustrated in FIG. **16**, C **1601** represents the DCT block to be up-sampled in the DCT domain, and c **1602** represents the equivalent block in the spatial domain. The two blocks are related to one another through the definition of the N-pt DCT and IDCT **1603**, see Rao and Yip, "Discrete Cosine Transform: Algorithms, Advantages and Applications," Academic, Boston, 1990. For convenience, the expressions are also given below.

The DCT definition is

$$C_q = z_q \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} c_i \cos\left(\frac{(2i+1)q\pi}{2N}\right), \text{ and} \tag{13}$$

the IDCT definition is

$$c_j = \sqrt{\frac{2}{N}} \sum_{q=0}^{N-1} z_q C_q \cos\left(\frac{(2j+1)q\pi}{2N}\right), \tag{14}$$

where

$$z_q = \begin{cases} \frac{1}{\sqrt{2}}; & q = 0 \\ 1; & q \neq 0 \end{cases}. \tag{15}$$

Given the above, block E **1610** represents the up-sampled DCT block based on filtering C with X_u **1611**, and e represents the up-sampled spatial domain block-based on filtering c with the x_u **1621** given by equation (12). Note that e and E are related through a 2N-pt DCT/IDCT **1630**. The input-output relations of the filtered input are given by,

$$E_k = \sum_{q=0}^{N-1} C_q X_u(k, q);\ 0 \le k \le 2N-1, \text{ and} \tag{16a}$$

$$e_i = \sum_{j=0}^{N-1} c_j x_u(i, j);\ 0 \le i \le N-1. \tag{16b}$$

As shown in FIG. **16**, the desired DCT blocks are denoted by A **1611** and B **1612**. The aim of this derivation is to derive filters X_{ca} **1641** and X_{cb} **1642** that can be used to compute A and B directly from C, respectively.

As the first step, equation (14) is substituted into equation (16b). The resulting expression is the spatial domain output e as a function of the DCT input C, which is given by,

$$e_i = \sum_{q=0}^{N-1} C_q \left[\sqrt{\frac{2}{N}}\, z_q \sum_{j=0}^{N-1} x_u(i, j) \cdot \cos\left(\frac{(2j+1)q\pi}{2N}\right) \right]. \tag{17}$$

To express A and B in terms of C using equation (17), the spatial domain relationship between a, b and e is

$$a_i e_i;\ 0 \le i \le N-1\ \ b_{i-N} e_i;\ N \le i \le 2N-1' \tag{18}$$

where i in the above denotes the spatial domain index. The DCT domain expression for a is given by,

$$A_k = z_k \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} a_i \cos\left(\frac{(2i+1)k\pi}{2N}\right). \tag{19}$$

Using equations (17)–(19) gives,

$$A_k = \sum_{q=0}^{N-1} C_q \left[\frac{2}{N} z_k z_q \sum_{i=0}^{N-1} \cos\left(\frac{(2i+1)k\pi}{2N}\right) \sum_{j=0}^{N-1} x_a(i, j) \cos\left(\frac{(2j+1)q\pi}{2N}\right) \right] \quad (20)$$

which is equivalently expressed as

$$A_k = \sum_{q=0}^{N-1} C_q X_{ca}(k, q) \quad (21)$$

where

$$X_{ca}(k, q) = \frac{2}{N} z_k z_q \sum_{i=0}^{N-1} \cos\left(\frac{(2i+1)k\pi}{2N}\right) \sum_{j=0}^{N-1} x_a(i, j) \cos\left(\frac{(2j+1)q\pi}{2N}\right). \quad (22)$$

Similarly,

$$B_k = \sum_{q=0}^{N-1} C_q \left[\frac{2}{N} z_k z_q \sum_{i=0}^{N-1} \cos\left(\frac{(2i+1)k\pi}{2N}\right) \sum_{j=0}^{N-1} x_a(i+N, j) \cos\left(\frac{(2j+1)q\pi}{2N}\right) \right] \quad (23)$$

which is equivalently expressed as

$$B_k = \sum_{q=0}^{N-1} C_q X_{cb}(k, q) \quad (24)$$

where

$$X_{cb}(k, q) = \frac{2}{N} z_k z_q \sum_{i=0}^{N-1} \cos\left(\frac{(2i+1)k\pi}{2N}\right) \sum_{j=0}^{N-1} x_a(i+N, j) \cos\left(\frac{(2j+1)q\pi}{2N}\right). \quad (25)$$

The above filters can then be used to up-sample a single block of a given dimension to a larger number of blocks, each having the same dimension as the original block. More generally, the filters derived here can be applied to any system that requires arithmetic operations on up-sampled DCT data.

To implement the filters given by equations (22) and (25), it is noted that each expression provides a kxq matrix of filter taps, where k is the index of an output pixel and q is the index of an input pixel. For 1D data, the output pixels are computed as a matrix multiplication. For 2D data, two steps are taken. First, the data is up-sampled in a first direction, e.g., horizontally. Then, the horizontally up-sampled data is up-sampled in the second direction, e.g., vertically. The order of direction for up-sampling can be reversed without having any impact on the results.

For horizontal up-sampling, each row in a block is operated on independently and treated as an N-dimensional input vector. Each input vector is filtered according to equations (21) and (24). The output of this process will be two standard DCT blocks.

For vertical up-sampling, each column is operated on independently and again treated as an N-dimensional input vector. As with the horizontal up-sampling, each input vector is filtered according to equations (21) and (24). The output of this process will be four standard DCT blocks as shown in FIG. 15C.

Syntax Conversion

As stated for the above applications of the transcoder according to the invention, one of the key applications for this invention is MPEG-2 to MPEG-4 conversion. Thus far, the focus is mainly on the architectures used for drift compensation when transcoding to a lower spatial resolution and additional techniques that support the conversion to lower spatial resolutions.

However, syntax conversion between standard coding schemes is another important issue. Because we believe that this has been described by patent applications already pending, we do not provide any further details on this part.

Although the invention has been described by way of examples of preferred embodiments, it is to be understood that various other adaptations and modifications can be made within the spirit and scope of the invention. Therefore, it is the object of the appended claims to cover all such variations and modifications as come within the true spirit and scope of the invention.

We claim:

1. A method for transcoding groups of macroblocks of a partially decoded input bitstream, the groups including intra-mode and inter-mode macroblocks, and each macroblock including DCT coefficients, and a motion vector, comprising:

mapping the modes of each group of macroblocks of the partially decoded input bitstream to be identical only if there is an inter-mode macroblock and an intra-mode macroblock in the group, and modifying the DCT coefficients and the motion vector in accordance with the mapping for each changed macroblock; and

down-sampling each group of macroblocks to generate reduced-resolution macroblock for an output compressed bit stream.

2. The method of claim 1 wherein the mode of each changed macroblock is mapped to inter-mode, and the motion vector and the DOT coefficients of each changed macroblock is set to zero if the partially decoded input bitstream has a relatively small amount of motion.

3. The method of claim 1 wherein the mode of each changed macroblock is mapped to inter-mode, and the motion vector of the changed block is predicted, and the DCT coefficients of the changed macroblock are converted to inter-mode if the bitstream has a relatively large amount of motion.

4. The method of claim 1 wherein the mode of each changed macroblock is mapped to intra-mode, and the motion vector of the changed macroblock is set to zero, and the DOT coefficients of the changed macroblock are converted to intra-mode if the bitstream has a relatively large amount of motion.

5. The method of claim 1 wherein the down-sampling includes mapping the motion vector to a low-resolution motion vector and further comprising:

variable length decoding a compressed bitstream to generate inverse DCT coefficients and the motion vector of the partially decoded bit stream;

inverse quantizing the inverse DCT coefficients using a first spatial quantizer to obtain the DOT coefficients;

quantizing each reduced-resolution macroblock using a second spatial quantizer; and

variable length coding each quantized reduced-resolution macroblock and the low-resolution motion vector.

6. The method of claim 1 further comprising:

generating a reduced-resolution drift-compensating signal for each down-sampled macroblock; and

adding the reduced-resolution drift-compensating signal to each down-sampled macroblock to compensate for drift in the output compressed bitstream.

7. The method of claim 1 further comprising:

generating a full-resolution drift compensating signal for each down-sampled macroblock;

adding each full-resolution drift-compensating signal to each macroblock of the group.

8. The method of claim 7 further comprising:

subtracting an inverse quantized signal and up-sampled signal from an original resolution reference signal to generate the full-resolution signal.

9. The method of claim 7 further comprising:

subtracting an inverse quantized signal from a reduced resolution reference signal; and

up-sampling the reduced resolution difference signal to generate the full-resolution signal.

10. The method of claim 7 further comprising:

subtracting an inverse quantized and up-sampled signal from an original resolution reference signal to generate the full-resolution signal.

11. The method of claim 1 further comprising:

generating a reduced-resolution difference signal for each down-sampled macroblock;

up-sampling each reduced-resolution difference signal to a full-resolution drift-compensation signal; and

adding each full-resolution drift-compensating signal to each macroblock of the group.

12. The method of claim 1 further comprising:

generating a full-resolution difference signal for each down-sampled macroblock; and

adding each full-resolution drift-compensating signal to each macroblock of the group.

13. The method of claim 1 wherein each macroblock includes $2^N \times 2^N$ pixels, and the down-sampling further comprises:

filtering the group of $2^N \times 2^N$ macroblocks to generate a single $2^N \times 2^N$ macroblock.

14. The method of claim 1 wherein the partially decoded input bitstream is in MPEG-2 format, and the compressed output bitstream is in MPEG-4 format.

15. The method of claim 1 wherein the transcoding is performed in a adaptive server of a multimedia content distribution system.

16. The method of claim 1 wherein the transcoding is performed in a transcoder of a home network.

17. The method of claim 1 further comprising:

applying a plurality of DOT filters to the DCT coefficients of each macroblock to generate a plurality of up-sampled macroblocks for each macroblock, there being one up-sampled macroblock generated by each filter, and where the macroblock and up-sampled macroblock has an identical dimensionality.

18. An apparatus for transcoding groups of macroblocks of a partially decoded input bitstream, the groups including intra-mode and inter-mode macroblocks, and each macroblock including DCT coefficients, and a motion vector, comprising:

means for mapping the modes of each group of macroblocks of the partially decoded input bitstream to be identical only if there is an inter-mode macroblock and an intra-mode macroblock in the group, and modifying the DCT coefficients and the motion vector in accordance with the mapping for each changed macroblock; and

means for down-sampling each group of macroblocks to generate reduced-resolution macroblock for an output compressed bitstream.

* * * * *